DTIC FILE COPY

AD-A199 830

DTIC ACCESSION NUMBER

PHOTOGRAPH THIS SHEET

LEVEL

INVENTORY

1

AFWAL-TR-88-2089

DOCUMENT IDENTIFICATION

OCT 1988

This document has been approved for public release and sale; its distribution is unlimited.

DISTRIBUTION STATEMENT

ACCESSION FOR		
NTIS	GRA&I	X
DTIC	TAB	
UNANNOUNCED		
JUSTIFICATION		
BY		
DISTRIBUTION /		
AVAILABILITY CODES		
DIST	AVAIL AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DTIC

SELECTED

NOV 01 1988

E

DATE ACCESSIONED

DATE RETURNED

88 11 01 062

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NO.

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDAC

DTIC FORM 70A DEC 83

DOCUMENT PROCESSING SHEET

PREVIOUS EDITION MAY BE USED UNTIL STOCK IS EXHAUSTED.

AFWAL-TR-88-2089

TIP CLEARANCE SIGNAL PROCESSOR DEVELOPMENT

Lawrence W. Langley

Vatell Corporation
P.O. Box 66
Christiansburg, VA 24073

October 1988

Final Report for Period July 1987 - May 1988

Approved for Public Release; Distribution is Unlimited

AERO PROPULSION LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6563

35 UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for Public Release; Distribution is Unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFWAL-TR-88-2089

6a. NAME OF PERFORMING ORGANIZATION
Vatell Corporation

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Aero Propulsion Laboratory (AFWAL/POTX)
Air Force Wright Aeronautical Laboratories

6c. ADDRESS (City, State, and ZIP Code)
P.O. Box 66
Christiansburg, VA 24073

7b. ADDRESS (City, State, and ZIP Code)
Wright-Patterson AFB, OH 45433-6563

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Air Force Systems Command

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F33615-87-C-2802

8c. ADDRESS (City, State, and ZIP Code)
Aeronautical Systems Div/PMRSA
Wright-Patterson AFB, OH 45433-6503

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
65502F	3005	20	85

11. TITLE *(Include Security Classification)*
Tip Clearance Signal Processor Development

12. PERSONAL AUTHOR(S)
Lawrence W. Langley

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 31Jul87 TO 24May88	1988 October	55

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Turbomachine Tip Clearance Signal Processing

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A signal processing system was designed and constructed to detect peaks of the Vatell eddy current clearance sensor signal, measure their amplitude and timing, and compute a correction for machine speed to indicate clearance and time of arrival for individual blades of a turbomachine. The system is based on an ADFIBM301 plug-in for the PC/XT, and consists of a second plug-in, along with software for signal detection and processing. Tests on a Pratt & Whitney JT15D first stage fan demonstrated timing precision of 0.16 microseconds and indicated individual blade clearances. The test results enable a projection of the capability to perform the speed correction calculation and indicate blade clearance continuously in real time.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Christopher J. Worland

22b. TELEPHONE *(Include Area Code)*
(513) 255-6802

22c. OFFICE SYMBOL
AFWAL/POTX

DD Form 1473, JUN 86 — *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

BACKGROUND INFORMATION

During the spring of 1986 Vatell Corporation developed a new concept for casing-based measurements of tip clearance, and obtained industrial sponsorship for some tests of this concept in a NASA-owned Pratt & Whitney JT15D engine, located at the Virginia Polytechnic Institute (VPI) Turbomachinery Laboratory in Blacksburg, Virginia. The tests showed that a clearance probe based on the new concept was feasible, and would produce high quality signals indicating both tip clearance and blade time-of-arrival. The tests also indicated that an important capability for jet engine monitoring and control could be developed around the Vatell sensor.

The signals produced by the sensor indicate clearance distance by amplitude, but signal amplitude is also a function of speed. However, the relationship between speed and amplitude is precise enough to allow direct compensation for speed using blade timing measurements as a source of speed information.

Figure 1 is a schematic view of the Vatell sensor which illustrates its principles of operation. The sensor contains two magnets, a flux bridge and a coil, all within a housing which may be filled with an encapsulant. The sensor is shown in typical orientation relative to a moving turbomachine blade made of electrically conductive material such as titanium.

One of the two magnets (1) is oriented with its North pole adjacent to a flux bridge. The second magnet (2) is oriented with its South pole adjacent to the flux bridge. The combination of the two magnets and the flux bridge produces a static magnetic field in the region traversed by the blade, adjacent to and between the South pole of magnet (1) and the North pole of magnet (2). The shape of this field and its extent are as described by Herbert C. Roters in Chapter 5 of his textbook "Electromagnetic

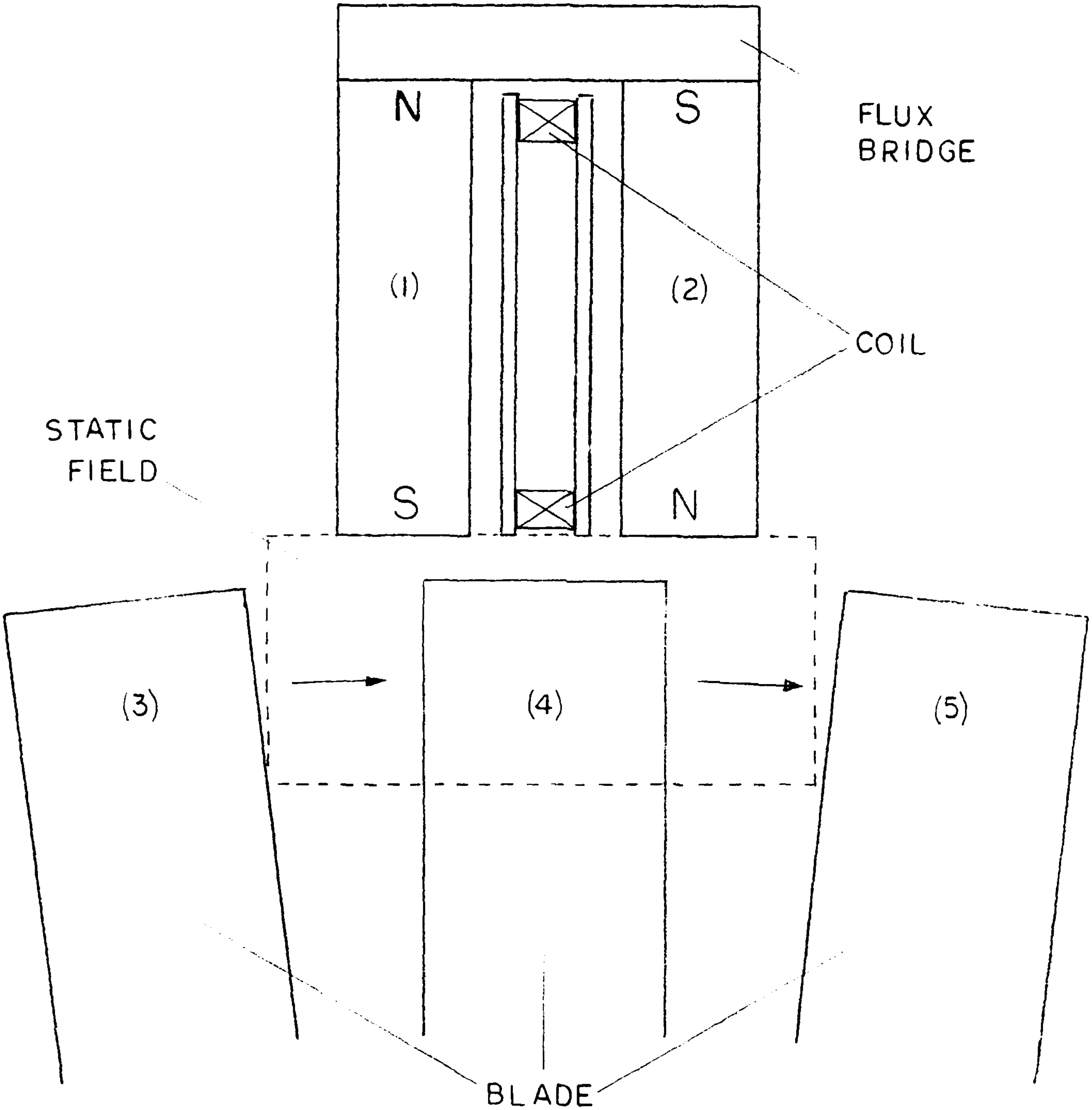


FIG. 1

Devices", published by Wiley. Figures 6a and 6b of the reference are particularly illustrative of the lines of force and the equipotential lines of the field of such a magnet. In the absence of any moving conductive objects in the region between the open poles of the two magnets, the field will not vary with time. Thus the field surrounding the coil will be constant, and no voltage will be produced at the terminals of the sensor. The effect of the blade on the static magnetic field, and the voltage produced by its motion in the coil, are best understood by visualizing that the blade is initially in a position (3) moving toward and through the position (4) to the position (5). As the blade begins to intercept field lines of the magnet, eddy currents will be induced in the conductive material of the blade. These currents will flow in the blade in a pattern and direction which oppose the increase in flux density passing through the blade material. The currents will effectively induce a magnetic field within the blade which is equal and opposite to that of the permanent magnet. External to the blade this field is effectively attached to the blade, and its motion relative to the coil will induce a voltage in the turns of the coil. This voltage is initially negative with the coil intercepting an increasing proportion of the eddy-current field. When the blade reaches the centered position (4) the eddy currents in the blade quickly reverse direction, because the field of the permanent magnet intercepted by the blade stops increasing and starts to decrease. The rapid reversal of eddy currents causes a rapid reversal of the induced field polarity seen by the coil, and a large positive peak voltage is produced because the field produced by these currents is closely coupled to the coil at this position. As the blade moves away from the centered position, the field intercepted by the nearest turns of the coil then decreases, and a second negative voltage peak is produced. It is negative because the polarity of the eddy currents has reversed and the blade is now moving away. The characteristic shape of the signal, shown in Figure 2 with polarity reversed, is the

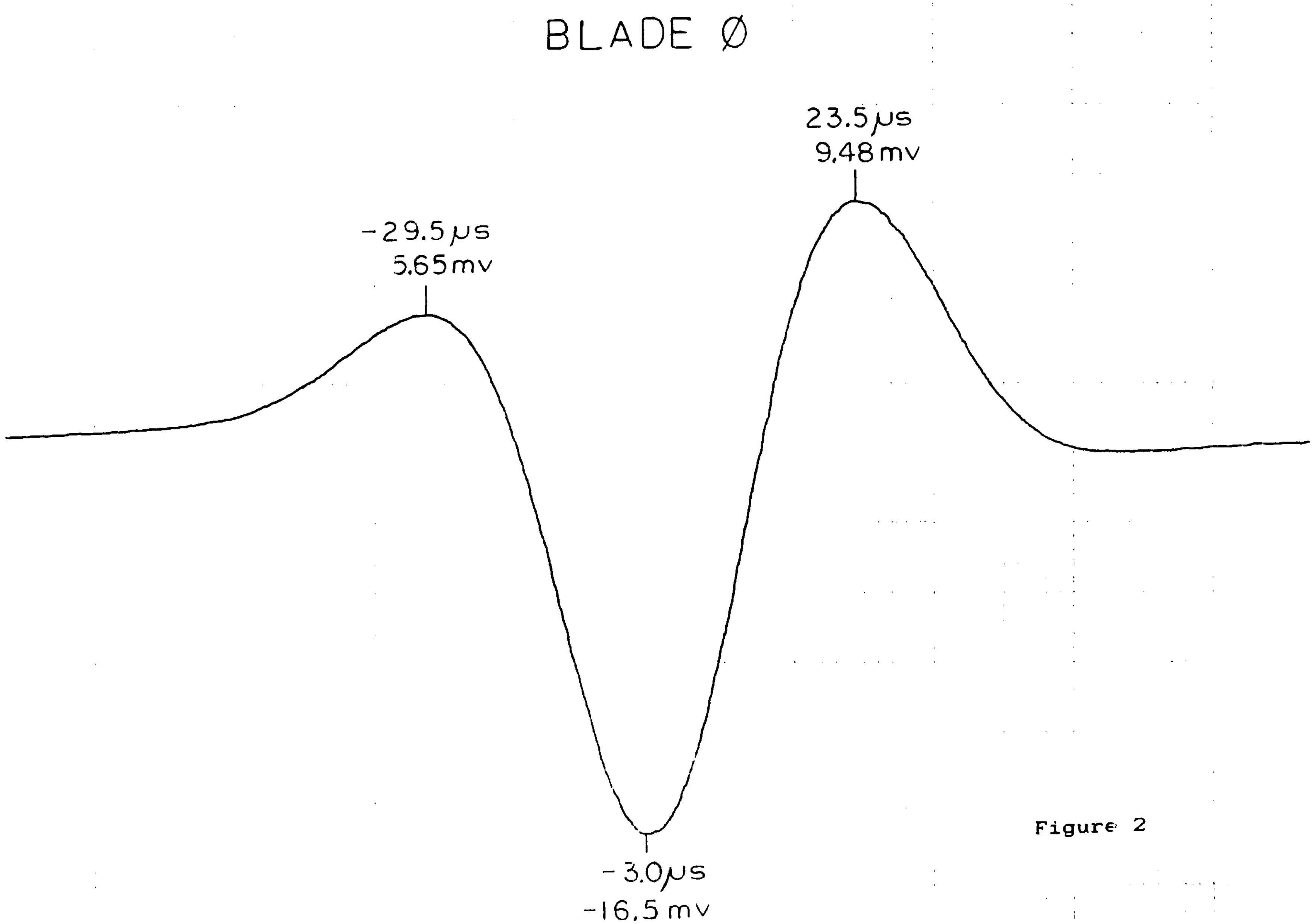


Figure 2

result of the growth and decay of eddy currents in the blade and the progression of changes in their orientation with respect to the coil.

Eddy currents in the moving blade are produced by the motion of the blade through the permanent magnet field, and the amplitude of the field they produce is proportional to the velocity of the blade. The voltages induced in the coil by these fields are also produced by relative motion, and are proportional to the blade velocity as well. The combination of these two proportionalities yields a square-law relationship between the signal level and the speed of the blade. The <u>shape</u> of the signal does not change with speed because the resistivity of the blade material is low and the eddy currents are dissipated only slightly by resistive losses. The inductance of the coil has a minimal effect because currents are induced in it by changes in the external field, rather than by voltages imposed on its terminals. The signal is therefore an almost perfect indicator of mechanical position. Its frequency content is much higher than the decay time constant of the driving eddy currents and much lower than the detection circuit's time constants.

Coupling of the permanent magnetic field to the blade, and the coupling of the coil to eddy-current induced transient fields, are both affected by the angle between the blade chord and the axis of the coil. For maximum signal, the plane of the sensor, which is the same as the plane of the coil, should be parallel to the blade chord. This orientation yields a maximum coupling between the permanent magnet and the blade material, and a maximum coupling between the coil and the eddy currents in the blade.

Coupling between the magnetic field and coil of the sensor and the blade material is also affected by the distance from blade to sensor. As the gap between them is increased, the signal induced by eddy currents will diminish. Thus the signal amplitude may be used for the measurement of blade clearance, if the effects of blade velocity can be compensated for.

Vatell submitted a 1987 SBIR proposal for development of 200°C rated sensor prototypes and for precise measurement of their characteristics in an actual jet engine compressor environment. The Air Force funded this work as contract F33615-87-C-2801. Measurements made during this contract showed the relationship between speed, clearance and signal amplitude to be expressed accurately by the equation:

$$\text{Amplitude(mv.)} = \frac{AS^2 - BS + C}{Y + D}$$

Where

S = Speed - RPM

Y = Clearance - Inches

The values of constants A, B, C and D for two sensors tested during this contract were:

Constant	Sensor #1	Sensor #4
A	1.03×10^{-8}	9.67×10^{-9}
B	4.89×10^{-5}	6.18×10^{-5}
C	0.075	0.159
D	0.0245	0.0220

To compensate a sensor for speed variation, the equation is solved for Y, thus:

$$\text{Clearance(in.)} = \frac{AS^2 - BS + C}{\text{Amplitude(mv.)}} - D$$

Using constants A, B, C and D for the sensor employed.

To make this sensor generally useful in tip clearance measurements, some means of carrying out the speed compensation, preferably on-line, is needed. Under the current contract Vatell has developed such a means, in the form of a breadboard signal processing module.

TECHNICAL OBJECTIVES

The principal objective of this contract was to design and build a breadboard signal processing module for use with the clearance sensors being developed under contract F33615-87-C-2801. Functions to be performed by the module, as listed in the 1987 SBIR Vatell proposal, were:

(1) Precise, wide-band differential amplification
(2) Conversion of signal amplitude to digital values
(3) Threshold detection
(4) Peak detection
(5) Timing of signal peaks
(6) Computation of machine speed
(7) Computation of clearance
(8) Communications with a host processor

Figure 3 is a block diagram of the processor from the Vatell proposal. The tentative organization for the module circuits was described as follows; signals from the sensor 9 would be amplified by a low-noise differential amplifier 5, for example a Dynamics Model 7525, to single-ended amplitude sufficient for high precision analog-to-digital conversion. This function would be performed by an A-D convertor 6, for example an Analogic ADAM 826-1 16-bit convertor with integral sample and hold, capable of converting the output of amplifier 5 to a 16 bit number in 2.3 microseconds. Conversion results would be read by the microprocessor 2, which also reads values of time from a clock 4. The software algorithm to be used for computation of clearance,

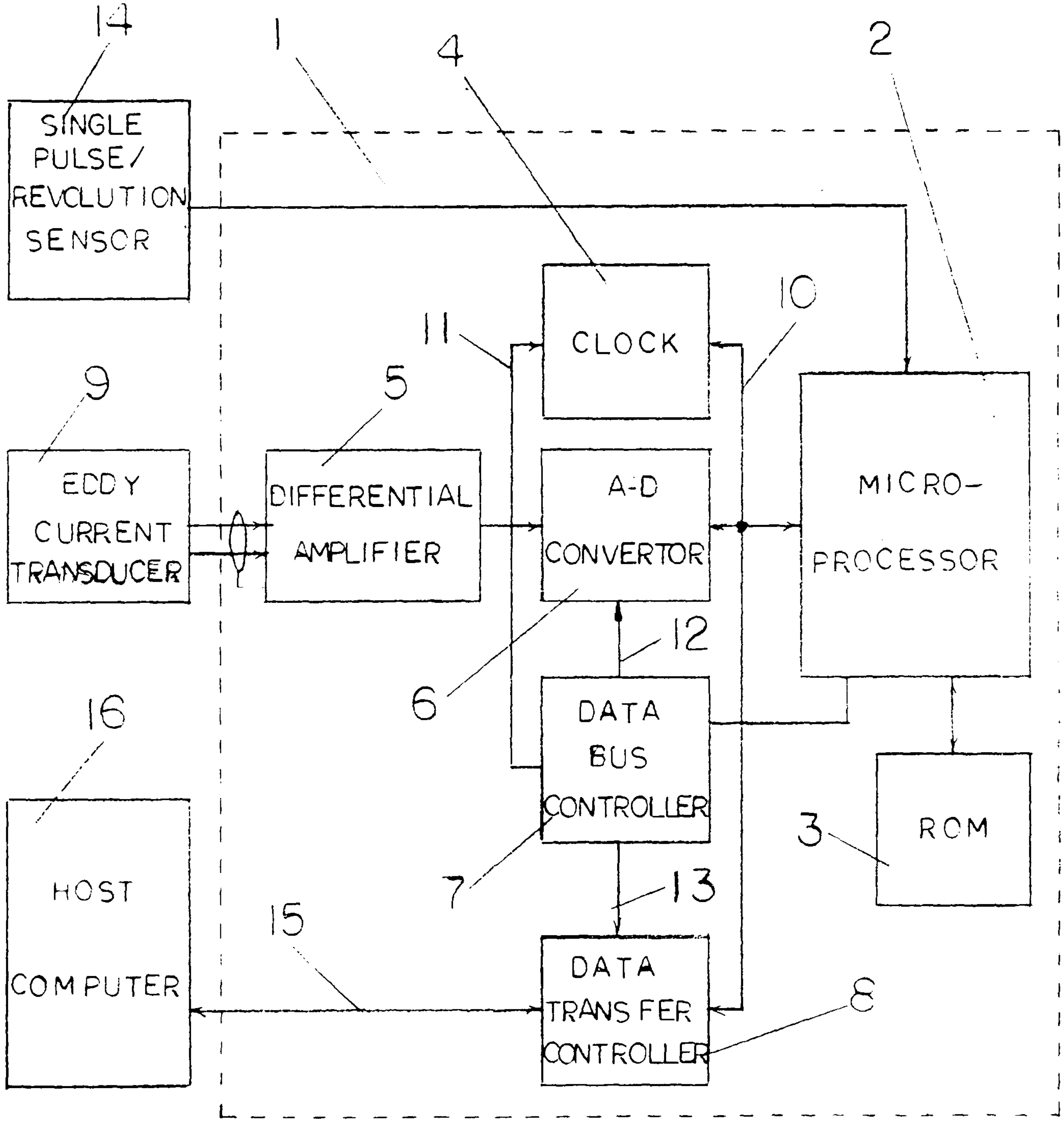

14
1
4
2
SINGLE PULSE/ REVOLUTION SENSOR
11
CLOCK
10
9
5
EDDY CURRENT TRANSDUCER
DIFFERENTIAL AMPLIFIER
A-D CONVERTOR
MICRO-PROCESSOR
12
6
16
DATA BUS CONTROLLER
ROM
3
HOST COMPUTER
7
13
15
DATA TRANSFER CONTROLLER
8

Figure 3

and the coefficients initially employed, were to be based on experimental test results for the clearance sensors being developed under the parallel sensor development contract. The coefficients (and, if necessary, the algorithm itself) would be modified as required. The end point of the computations would be:

(1) a clearance value for the blade which has just passed the sensor;
(2) a time value for the signal peak from this blade;
(3) the value for machine speed used in computing clearance; and
(4) the current blade number.

These values would all be computed in the time period between blade signal peaks and transferred during this period to a host computer. The host computer would store computed blade values for the current revolution, and compute whatever statistics of clearance, timing or other values are desired. Determination of these host computer functions, and their programming, were not to be included in this project.

Operation of the processor would be verified using signals from a sensor on the first stage fan of the JT15D at the VPI airport. This fan is 21 inches in diameter, has 28 blades, and operates at a maximum speed of approximately 16,000 rpm. A once-per-revolution sensor had been fitted to this engine as part of the work on the sensor contract.

PRELIMINARY DESIGN

A tentative selection of the Texas Instruments TMS 32010 Digital Signal Processor chip had been made at the time of the proposal. This chip was not the fastest or most powerful available, but was the only one being widely used by board-level sub-system suppliers. One of these suppliers, Atlanta Signal Processing Incorporated, offered a signal processing plug-in board for the IBM PC/XT based on this chip. This plug-in, the ADPIBM301, was designed to perform some of the functions required for this contract. Atlanta Signal Processing also offered design and debugging tools for TMS32010 program development. The ADPIBM301 was selected as the base for the Vatell signal processing module.

During the initial design calculations, it was determined that peak timing resolution would be severely limited if the original concept of software peak detection was used. That concept required that the analog sensor signal waveform be converted at the highest possible rate to a sequence of digital values. Each value would then be compared to the preceding one by software. in a subtract (or compare) and branch loop. The time for the minimum number of instructions to do this, added to an A/D conversion time, would yield a time precision of no better than 3 microseconds. Because one of the proposed applications for the signal processing module would be to measure blade time of passage for vibration and twist sensing, it was decided that this precision would not be acceptible. Instead, a hardware peak sensing scheme was devised and substituted for the software approach. This change had some side benefits: the A/D conversion cycle timing requirement could be relaxed, and a sample and hold circuit could be used to improve the precision of sample timing even further.

In the 1987 SBIR proposal Work Plan it was pointed out that because of the short duration and limited funding of the project the signal processing module would have to be developed using some commercially available parts. One of the parts originally selected was the Dynamics Model 7525 differential amplifier: it was proposed to amplify the balanced sensor signal to an A/D conversion voltage level. This amplifier is only available in a packaged unit designed for its own special housing; the total outline dimensions of the housing are greater than those of the PC/XT. Early tests of the new sensor indicated that its voltage output would be at least ten times greater than that of the original sensor. With this new sensor signal a very low noise, high gain amplifier would not be required. It was decided to substitute an IC differential amplifier on the module board for the Model 7525.

As a result of these decisions; i.e. to employ the Atlanta Signal Processor ADPIBM310 board, to use an on-board IC differential amplifier instead of the Dynamics unit, and to substitute a hardware peak finding and sampling method for the software method, it became possible to fit all the required signal processing functions into a single processor board. The decision to make this board a PC/XT plug-in was a natural result. The configuration finally adopted was;

(1) a Victor PC2 microcomputer with video monitor;

(2) a slightly modified ADPIBM310 board in one host computer slot, containing the TMS 32010 signal processing chip, A/D converter and memory; and

(3) a Vatell-designed signal acquisition board in the adjacent slot, containing differential amplifier, peak detection, timing and control circuits.

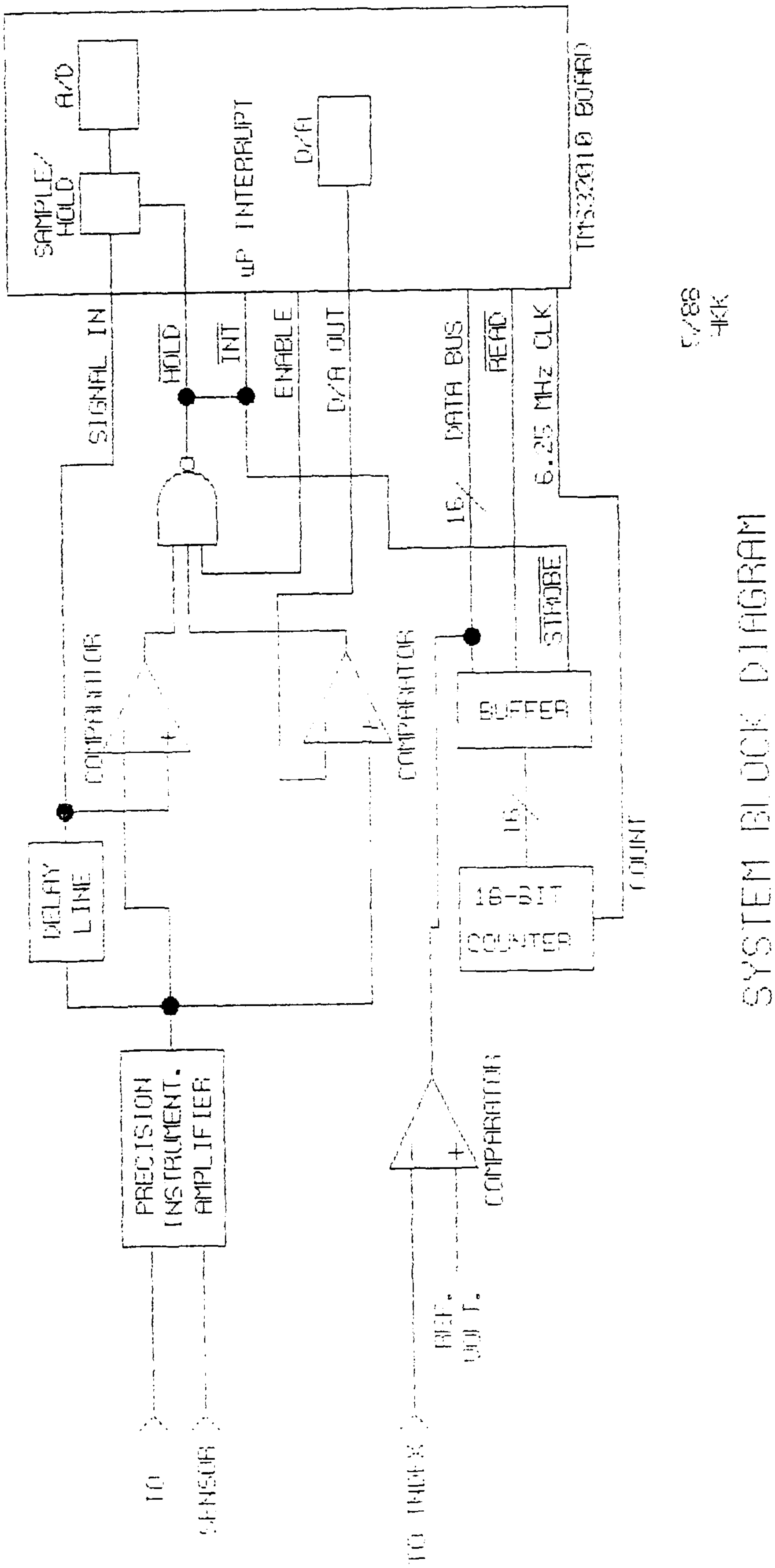
The two boards would be cross-connected by a ribbon cable, which would allow rapid and controlled data transfer between them. With this configuration it was not necessary to design a communications link between the PC and the module; software could be written for the PC to directly access module memory on the PC bus, and the Atlanta Signal Processing debugger could be used to control and test the system in operation. The detailed system and circuit designs were carried out on this basis.

DETAILED DESIGN

Figure 4 is a block diagram of the circuits on the signal acquisition board. The sensor signal is amplified to a proper level (0-5 volts dc) for analog-to-digital conversion, by a precision differential input amplifier. The output of the precision amplifier drives the two inputs of a signal comparator, one directly and the other through a 0.5 microsecond delay line. With this arrangement, every time the sensor signal reaches a peak (positive or negative) the output of the signal comparator reverses. The reversal of signal comparator output from positive to negative indicates that a positive peak has occurred in the sensor signal. To prevent this circuit from triggering analog to digital conversions on other peaks of the waveform or on noise, the signal comparator output is gated by another comparator. This second comparator compares the signal to a threshold (D/A) voltage, and only allows triggering of A/D conversion on a peak above the threshold. The threshold is set by software and can be adjusted dynamically for the best compromise between peak detector dynamic range and noise immunity.

When a positive peak in the sensor signal is detected, the analog voltage value of the delayed signal is captured by the ADPIBM310 sample and hold circuit, which has an aperture delay and uncertainty of less than 0.1µsec. At the same time the number in a continuously running 18-bit timing counter is transferred to a buffer. This number is the peak occurrence time; with a counting rate of 6.25 mhz it has a precision of 0.16 µsec. The A/D conversion cycle by the ADPIBM310 is also commanded at this time. The ADPIBM310 board then increments the blade counter and performs its calculations.

Figure 4 also shows that the module receives the once-per-revolution (index) signal. This signal is derived from a photoelectric detector which scans a retroreflective paint strip



SYSTEM BLOCK DIAGRAM
TMS32010 BOARD
A/D
SAMPLE/ HOLD
μP INTERRUPT
D/A
SIGNAL IN
HOLD
INT
ENABLE
D/A OUT
DATA BUS
READ
6.25 MHz CLK
16
STROBE
BUFFER
16
COUNT
16-BIT COUNTER
COMPARATOR
COMPARATOR
COMPARATOR
DELAY LINE
PRECISION INSTRUMENT. AMPLIFIER
TO SENSOR
TO INDEX
REF. VOLT.
5/88

Figure 4

on the engine spinner. The once-per-revolution signal is used to indicate the space between Blades 28 and 1; the signal itself is not used directly in blade timing. When the blade number is incremented by the ADPIBM310, the once-per-revolution signal is checked to determine if it has made a negative-going transition since the last blade count. If so, the blade count is reset to 1. The blade timing is <u>indicated</u> as 0, but the continuously running counter is not reset. The next and all succeeding values for blade timing are obtained by subtracting the old timer value from the new, then adding the difference to the count for the preceding blade. Thus all blade timing information is referenced to blade 1, but not to the once-per-revolution index. This is done because the timing of the once-per-revolution index may vary with engine speed, even though it always occurs between blades 28 and 1. Figure 5 is an oscillogram of the once-per-revolution signal. A photograph of the completed Vatell signal acquisition module is given as Figure 6. Figure 7 shows both boards installed in the Victor PC2.

The ADPIBM310 board is designed for processing of continuous waveforms such as audio signals, using FFT or digital filtering algorithms. Its analog to digital convertor is controlled to sample the signal waveform at regular intervals, with software selection of the sampling rate. The processing of signals from the Vatell sensor requires a different mode of control. Analog to digital conversion must be triggered at a precise time <u>relative to the waveform</u>, under control of the peak sensing circuit. To achieve this it was necessary to modify the A/D control circuits. Figure 8 is a photograph of the ADPIBM310 board which shows this modification.

One of the most important selection criteria for the Texas Instruments TMS 32010 Digital Signal Processor chip, and the ADPIBM310 board, was that the 32010 chip instruction set included a divide primitive instruction which could be used to build a

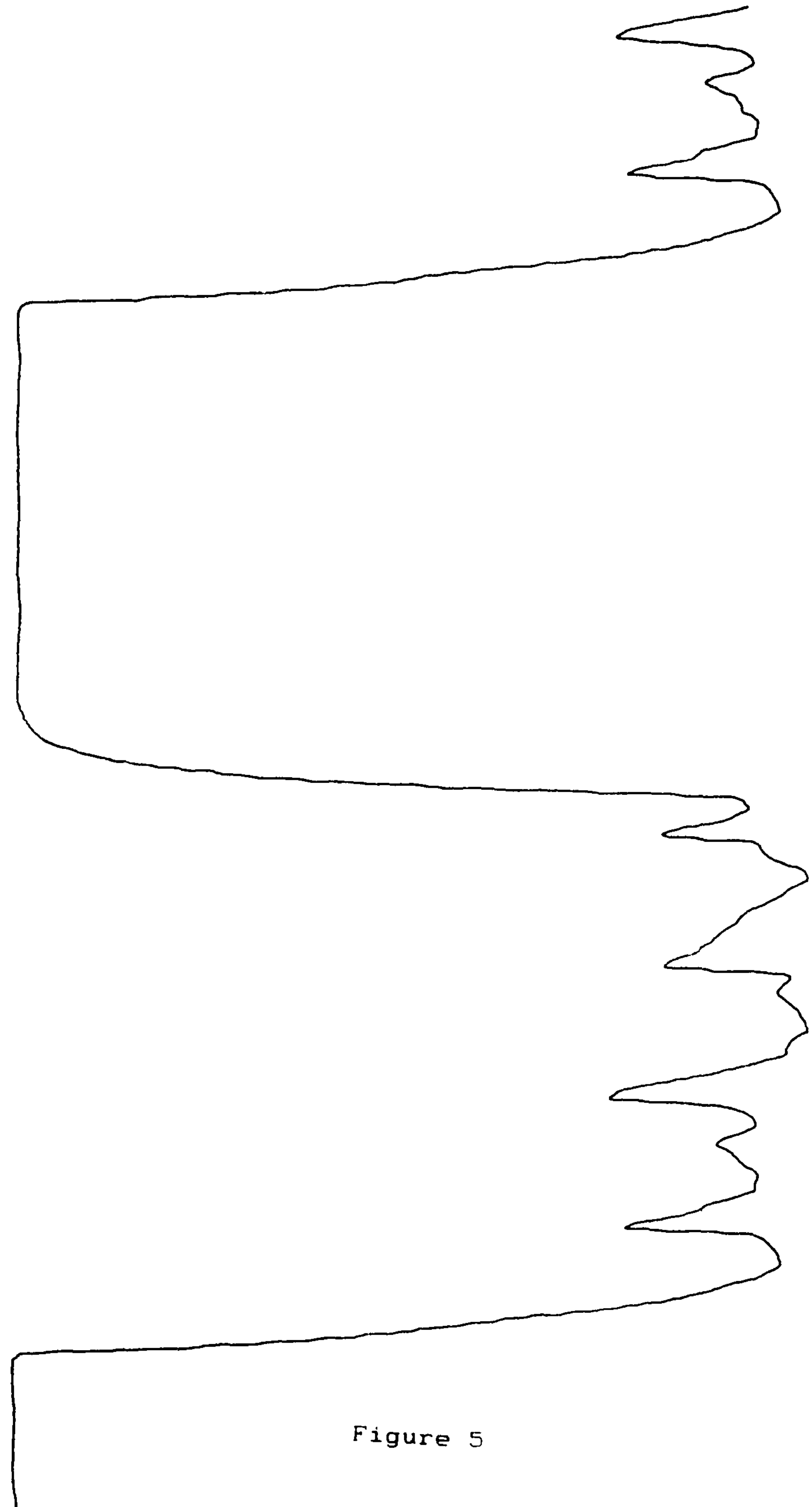


Figure 5

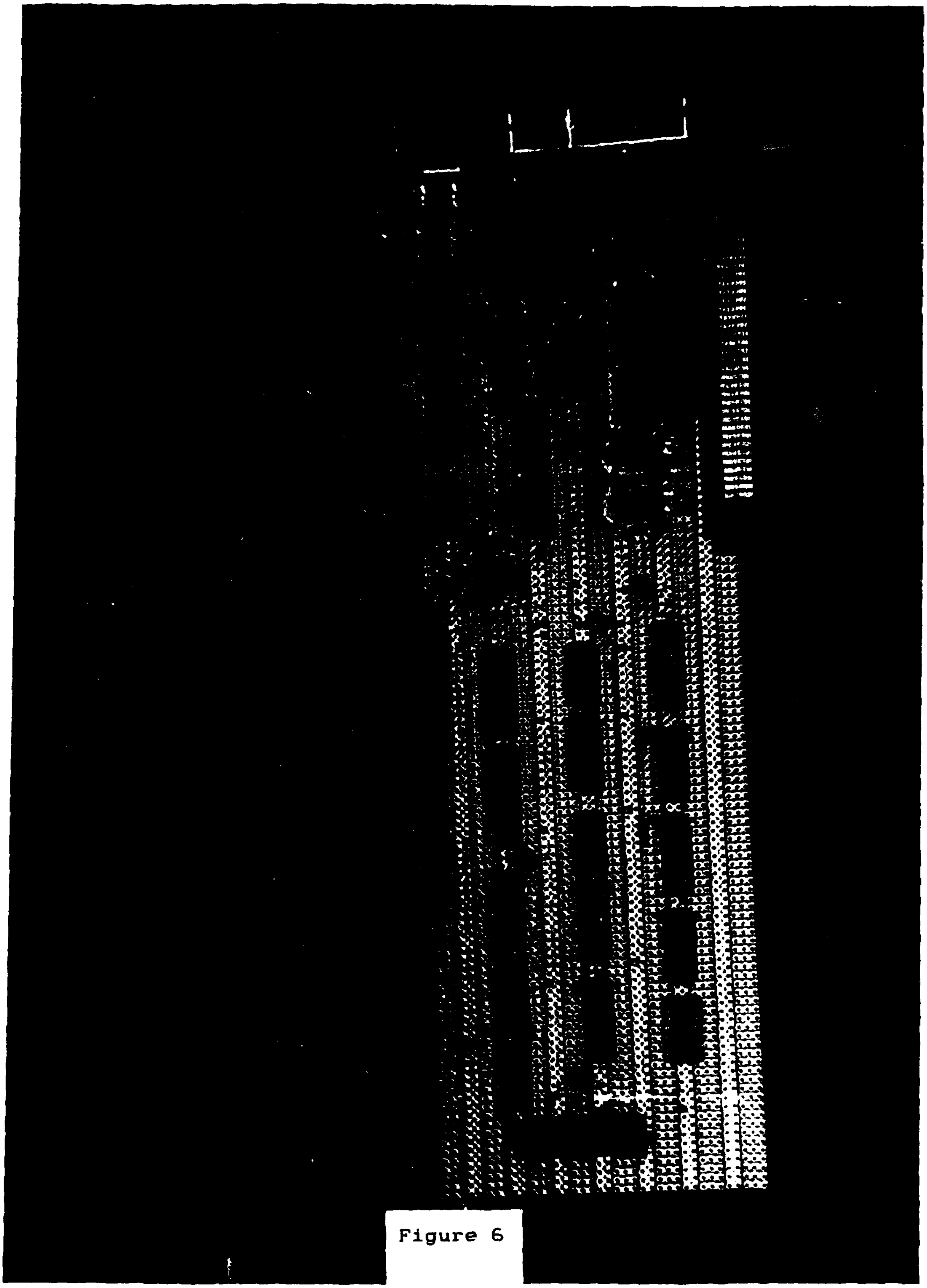


Figure 6

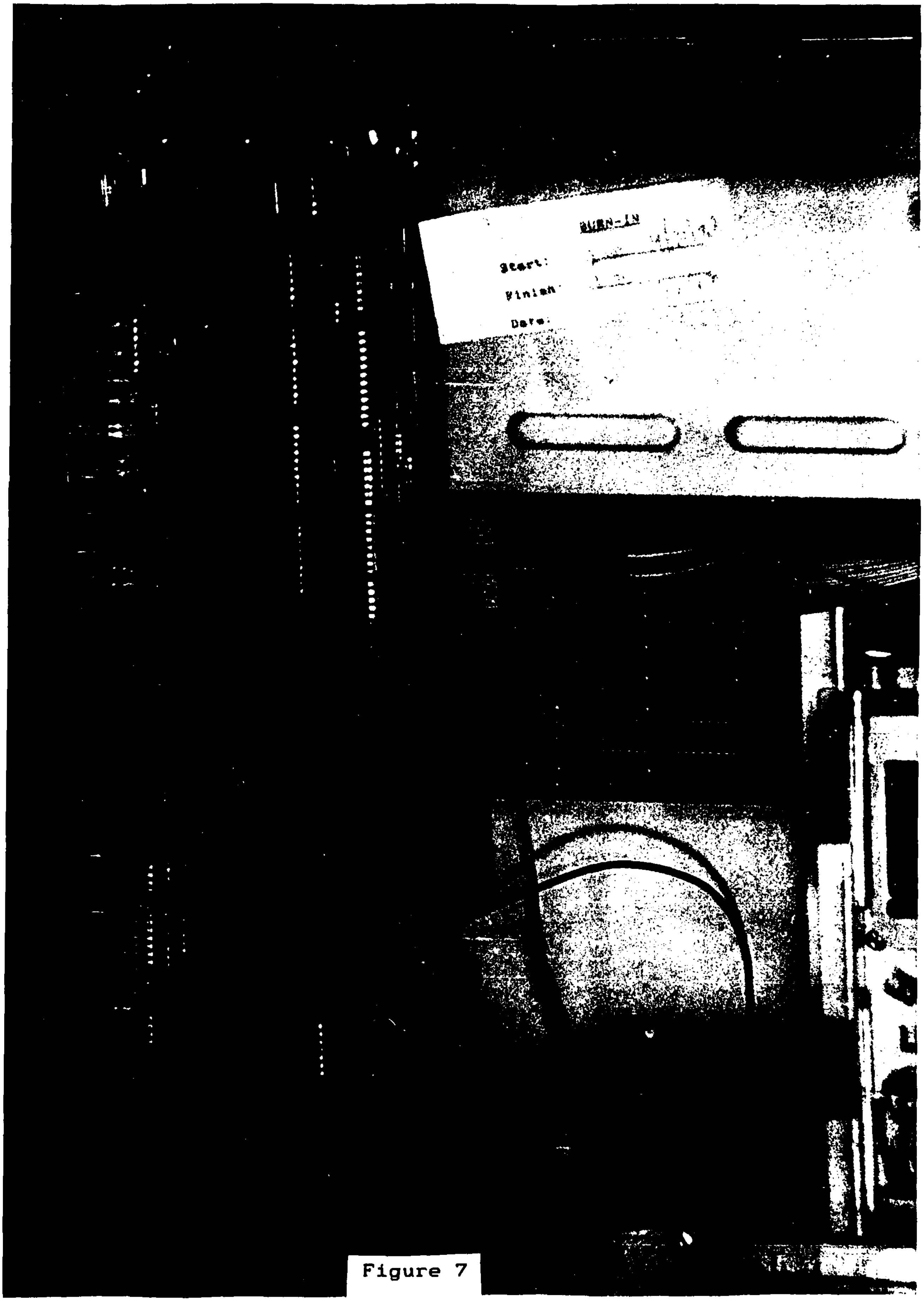


Figure 7

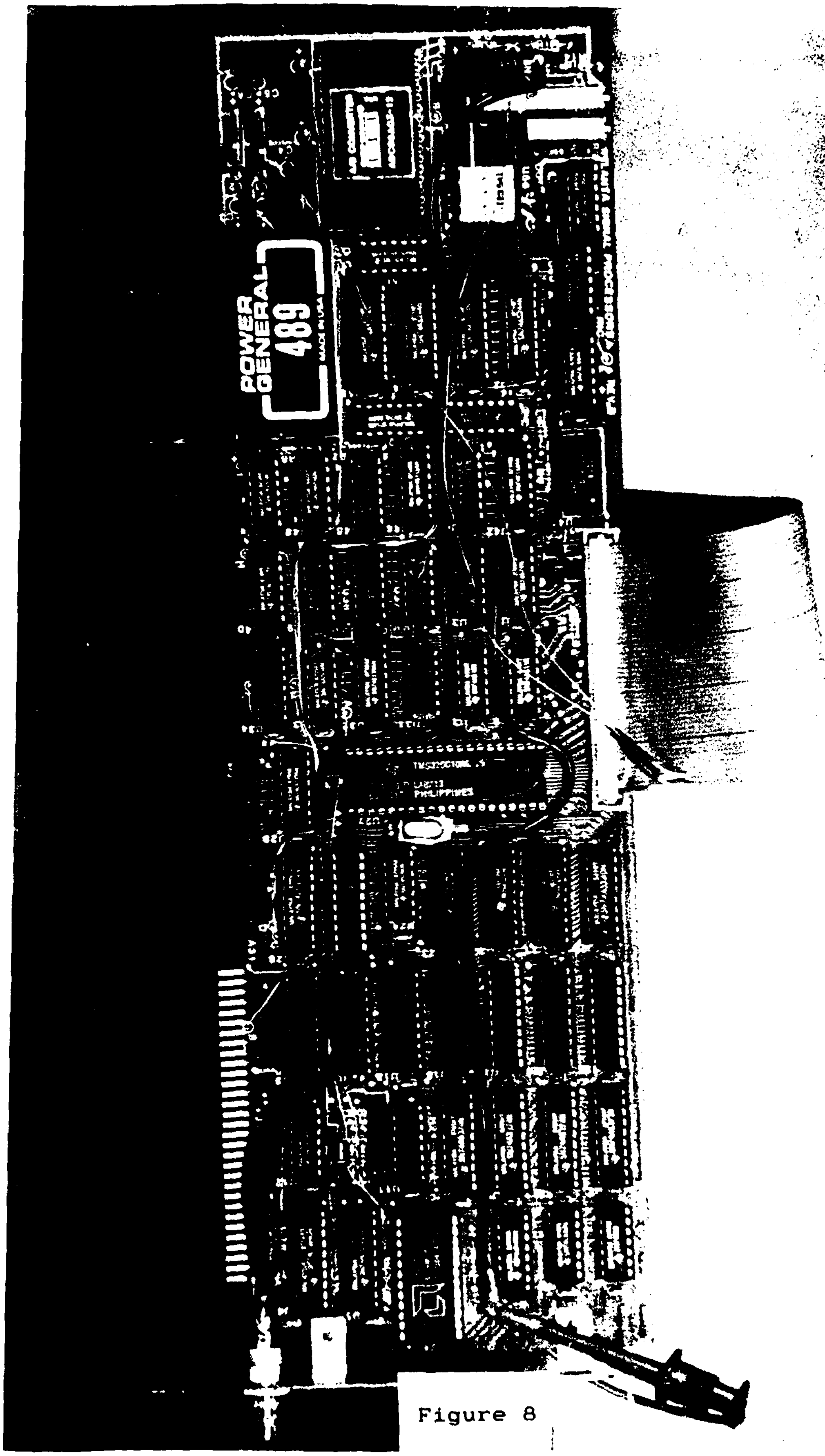


Figure 8

double precision divide macro. Vatell was led to believe that this would be a straightforward process, and actual writing of the macro was left to a late stage in the programming. The macro writing task was found to be a great deal more difficult than originally anticipated. Vatell contacted the supplier for help, and was told that no one had ever written a double precision divide for this chip. It would, indeed, be a very difficult task, and TI would not even quote on doing it. Furthermore, the TMS 32010 chip was "no longer supported" for program development, having been replaced by more advanced, faster hardware.

The only possible recourse to an expensive, time consuming replacement of the TMS 32010 was to perform the speed compensation calculations in the host computer. The amplitude and timing of peaks for a full revolution would be read by the host computer, which would then calculate clearances for all 28 peaks, using a Basic routine. Of course this could not be done in real time. Software for the TMS 32010 and the PC were completed according to this modified plan. Appendix A is a listing of the final TMS 32010 signal acquisition software and the host Basic routine for speed compensation.

SYSTEM TESTS

After bench tests of the signal acquisition board, the ADPIBM310 board and the host computer, the system was assembled and installed in the JT15D test stand at VPI and connected to the clearance and once-per-revolution sensors. The experimental installation is shown in Figure 9. The PC is at the far end of the engine control panel, together with the once-per-revolution sensor power supply and load circuits, which are housed in an aluminum box on the shelf. Figure 10 shows the engine mounted on its test stand, and Figure 11 shows the once-per-revolution sensor mounted adjacent to the first stage fan on the engine housing.

A number of minor problems were encountered during the tests. Hysteresis of the comparator used for peak detection was too great, and had to be reduced. Several software errors were found and corrected. The noise level on the amplified sensor signal was found to be excessive. While the signal entering the board was extremely clean, the high frequency background noise caused by operation of digital circuits on the ADPIBM310 and the signal acquisition board was almost 50 millivolts peak-to-peak at the output of the final stage of amplification, more than 1% of the peak signal. After a thorough review of the signal acquisition board layout and circuits, the following modifications were made:

(1) power supply lines on the board were divided into analog and digital groups, with separate connections back to the power source on the board;

(2) an RC bypass network was added to each leg of the first stage amplifier IC supply; and

(3) the bandwidth of the second stage of amplification was limited to 300 khz by addition of a filter network.

While these measures reduced noise by 1-2 db., it was still felt to be excessive. Reducing noise to an acceptable level (less than 5 mv.) would necessitate a completely new layout on a multi-layer board with separate analog and digital ground planes, and would probably require complete electrostatic and magnetic shielding of the analog circuits. At this point in the project, VPI's schedule for use of the JT15D called for it to be removed from service for major modifications in a few days. It was decided to complete the module tests without any further noise reduction efforts, to try to prove its capabilities without a costly and time-consuming major re-design.

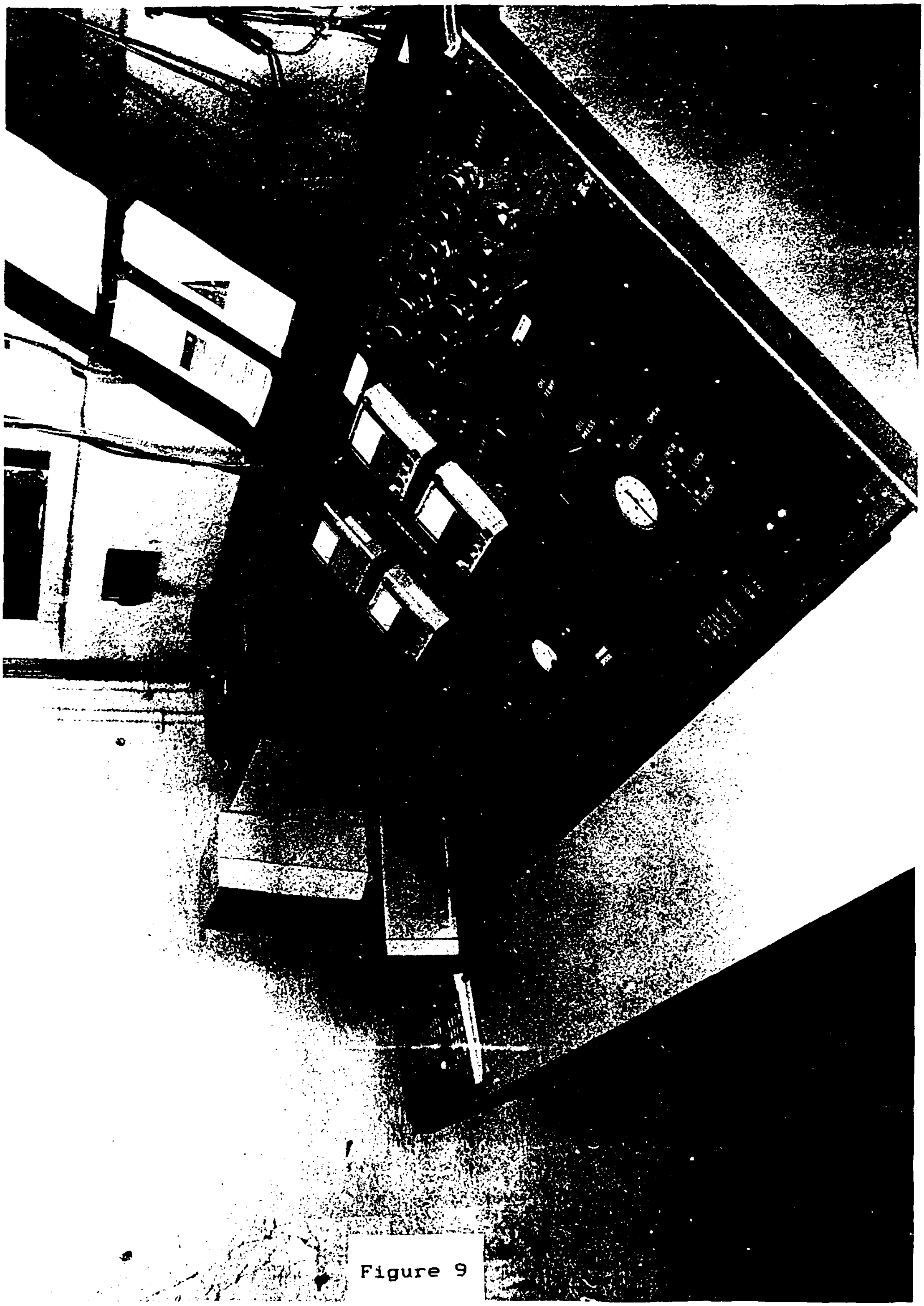


Figure 9

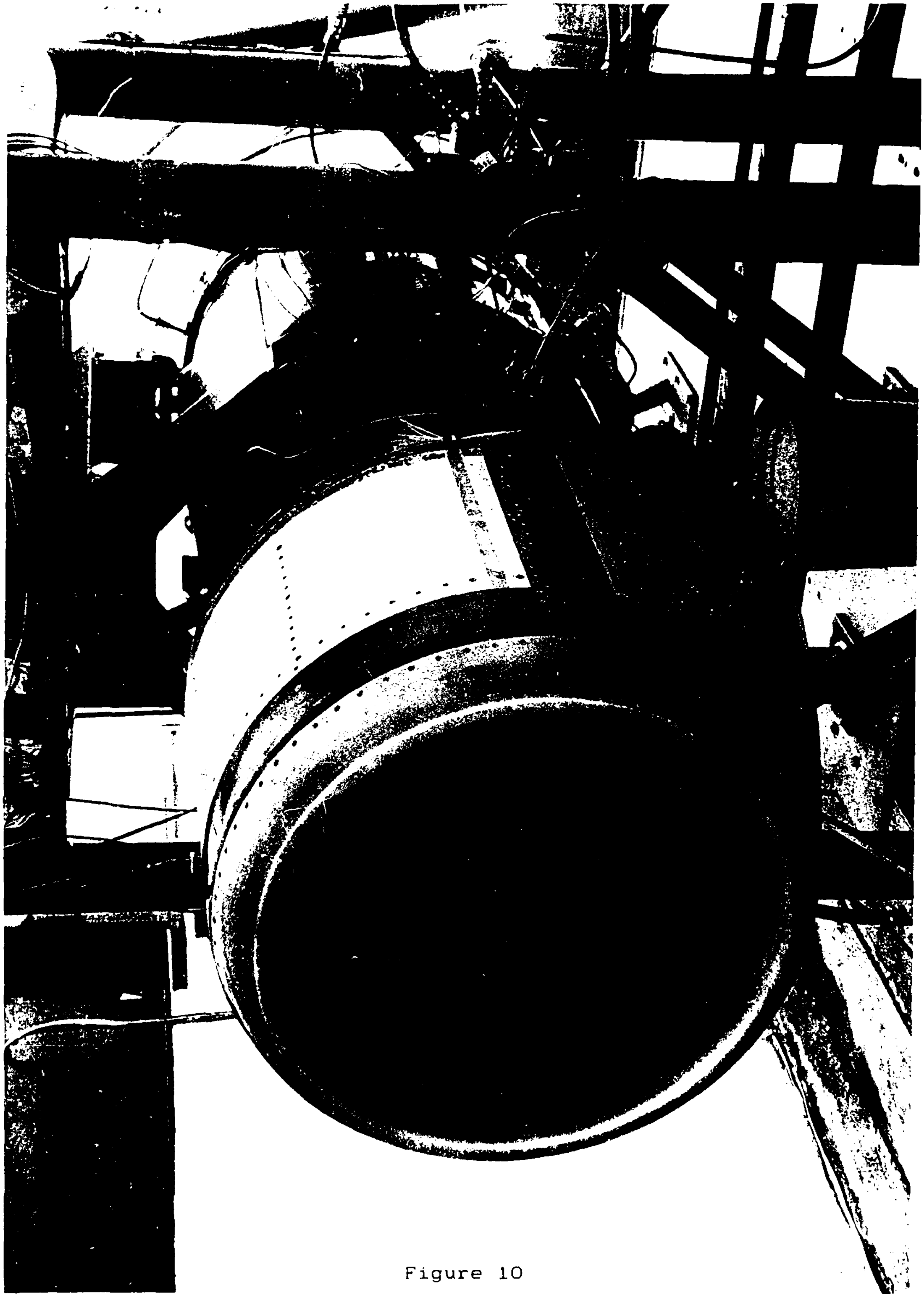


Figure 10

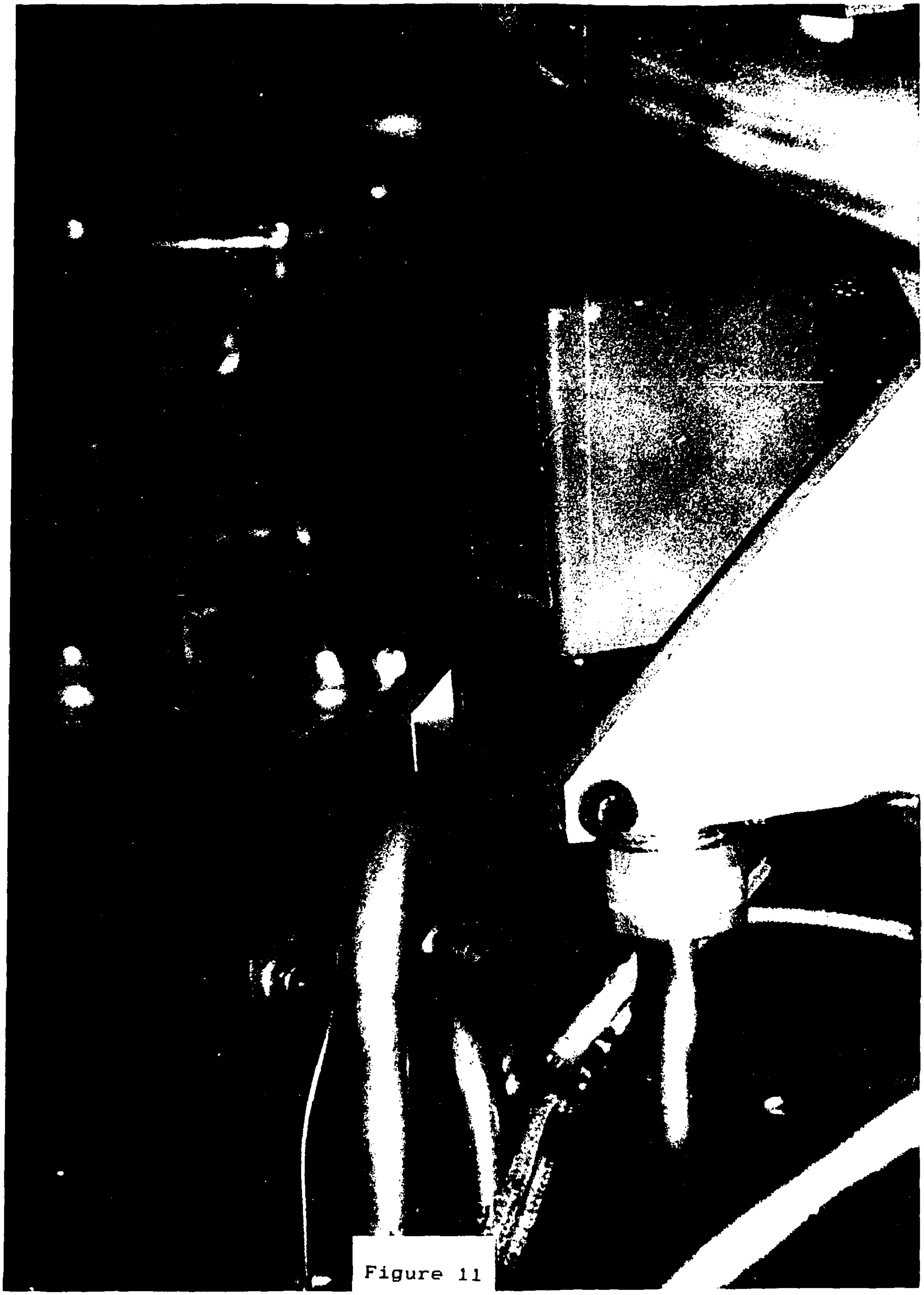


Figure 11

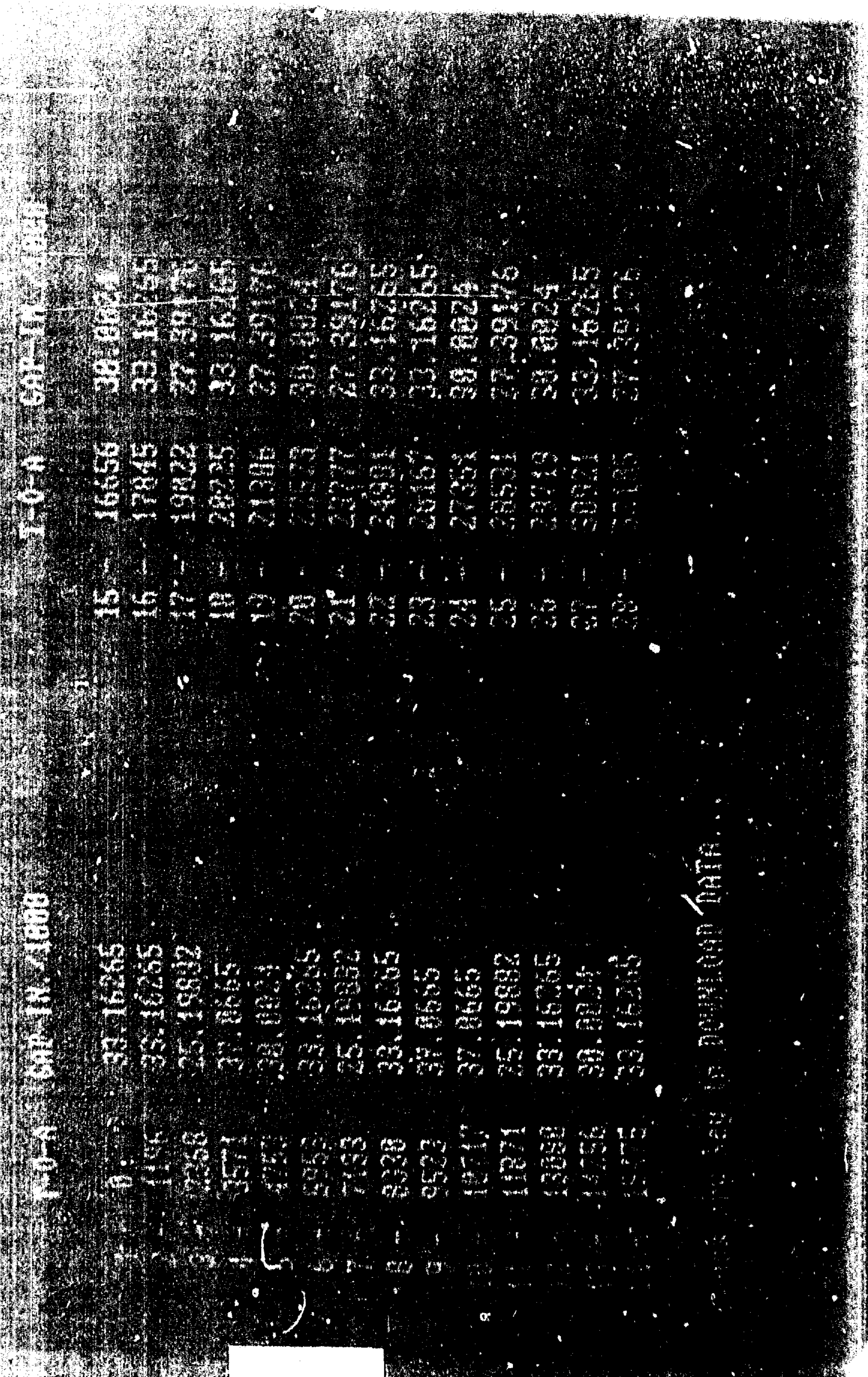

T-O-A GAP-IN./1000
33.16265
33.16265
25.19882
37.0665
38.0024
33.16265
25.19882
33.16265
37.0665
10717 37.0665
11871 25.19882
13088 33.16265
14256 30.0024
33.16265
T-O-A GAP-IN./1000
15 - 16656 30.0024
16 - 17845 33.16265
17 - 19022
18 - 20225 33.16265
19 - 21386 27.39176
20 - 30.0024
21 - 23777 27.39176
22 - 24981 33.16265
23 - 33.16265
24 - 27351 30.0024
25 - 27.39176
26 - 29719 30.0024
27 - 30921 33.16265
27.39176
DOWNLOAD DATA...

Figure 12

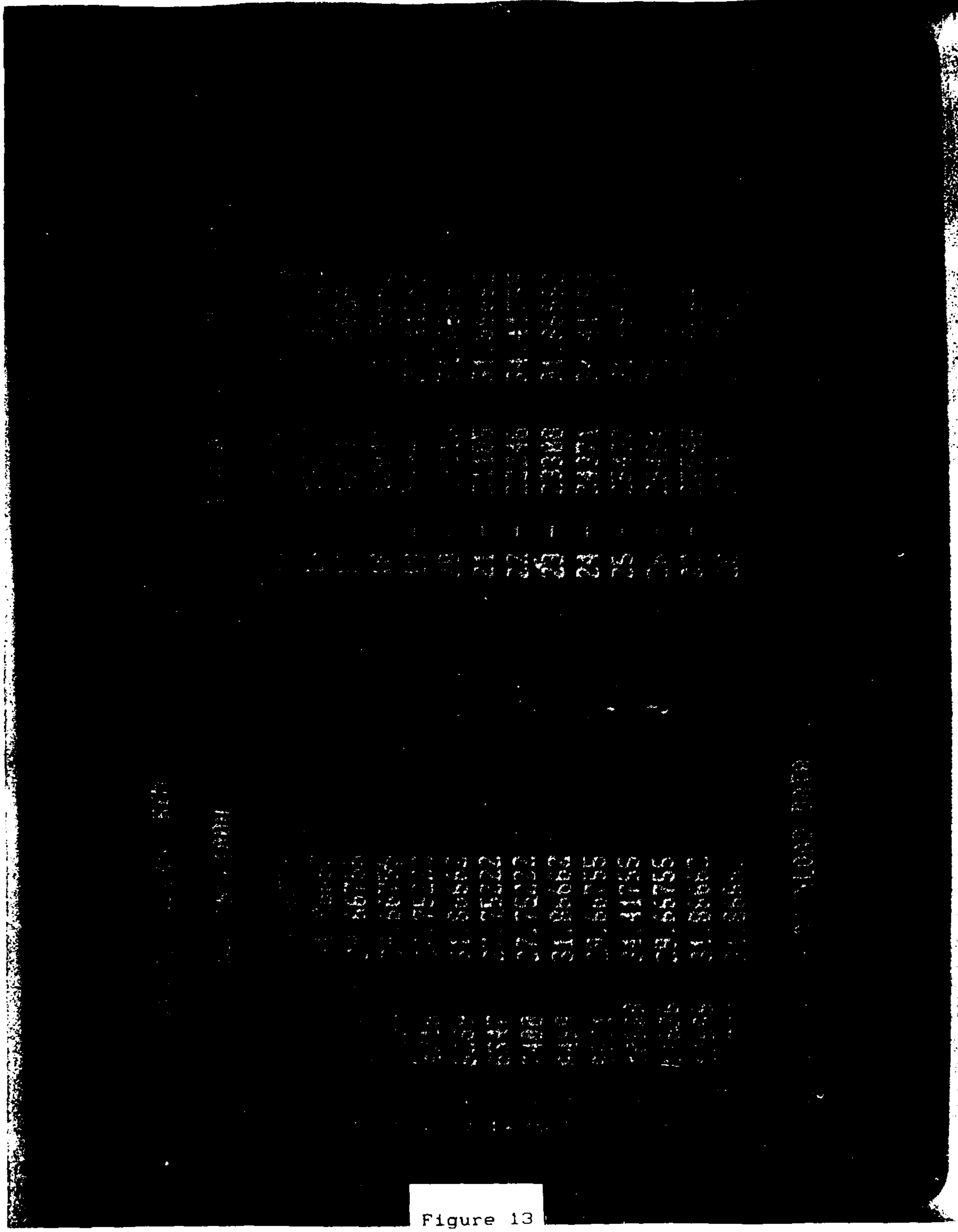


Figure 13

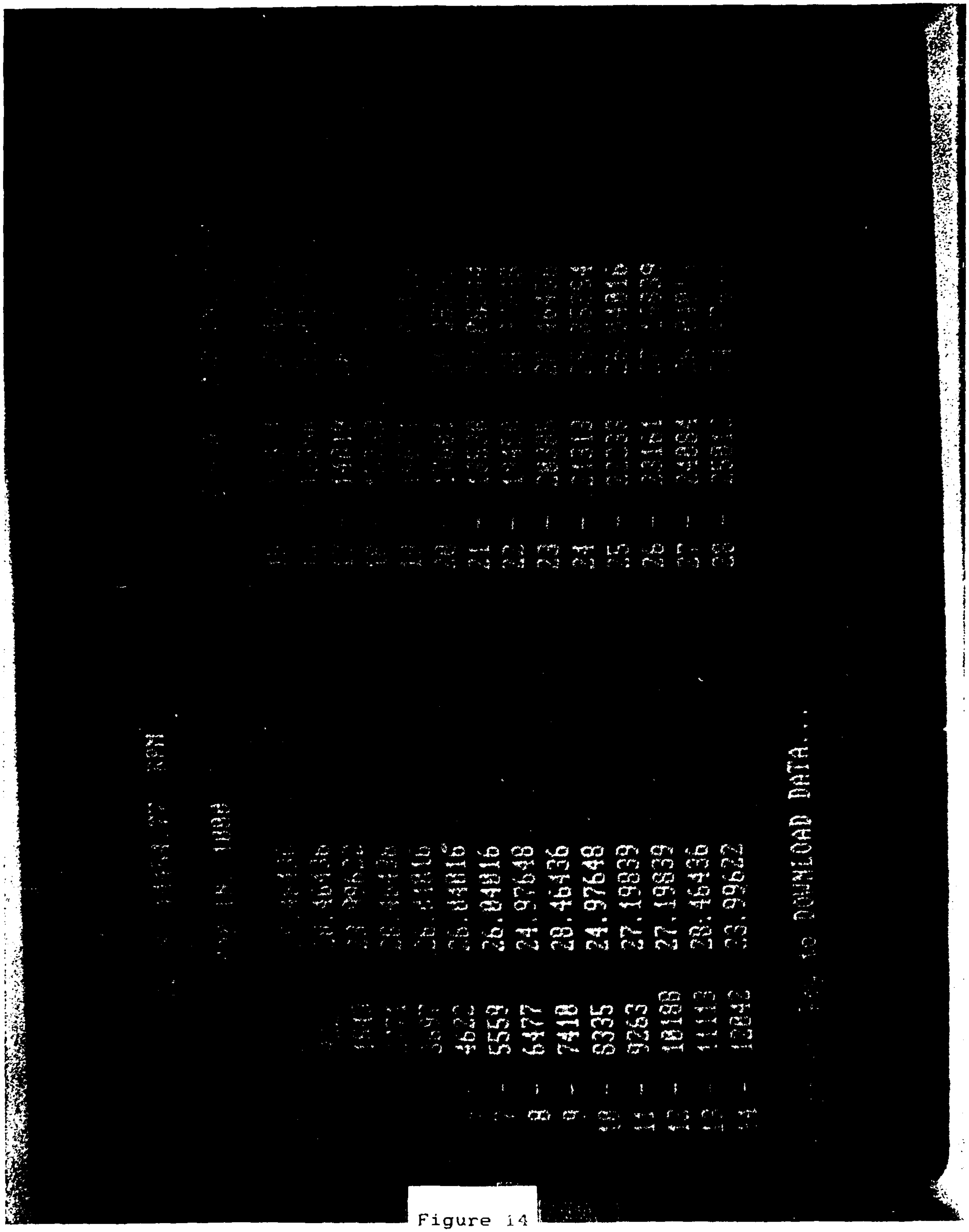


Figure 14

Figures 12, 13 and 14 show the monitor displays resulting from the final module tests, at three different engine speeds. The calculated speed is displayed at the upper left of the screen. The time value for blade 1 is indicated as 0, and all other blade times are referenced to this value. The least count for blade timing is equal to 0.16 μseconds. This is equivalent to a tangential displacement of approximately 0.002 inches at a speed of 15,000 rpm. Clearance values were calculated by the host computer using the following constants:

$A = 1.2 \times 10^{-8}$	$B = -1 \times 10^{-4}$
$C = .31$	$D = .02$

In addition a scaling multiplier for amplitude, $K_1 = 7 \times 10^{-5}$, was used to adjust for sensor gain, amplifier gain and analog to digital conversion gain. The factors A, B, C and D used in these tests were different from those calculated for the sensor. Because time ran out on the engine test, there was no opportunity to correct these values.

While the clearance calculation was performed off-line, the results were the same as they would have been if the calculation had been done by the TMS32010 during the interval between blades. The TMS32010 program for acquiring data used 122 instructions, executing in approximately 30-32 microseconds. There is time for execution of approximately 400 instructions during the interval between blades at the highest speed. Thus the concept of calculating clearance in real time appears to be viable.

ANALYSIS OF THE DATA

Tables 1, 2 and 3 show the results of comparisons between the times of arrival actually recorded and "ideal" times, for the three speeds tested. "Ideal" times were calculated by dividing the Blade 28 arrival time by 27, and then multiplying by the

Analysis of Vatell Sensor Output - Signal Processor Data

Turbine Speed is 11,263.97 RPM

Blade Number	T-O-A	Ideal Time	Time Error	Delta Time
1	0			
2	1196	1189	7	1196
3	2368	2378	-10	1172
4	3571	3567	4	1203
5	4753	4756	-3	1182
6	5953	5945	8	1200
7	7133	7134	-1	1180
8	8330	8324	6	1197
9	9523	9513	10	1193
10	10717	10702	15	1194
11	11871	11891	-20	1154
12	13080	13080	0	1209
13	14256	14269	-13	1176
14	15475	15458	17	1219
15	16656	16647	9	1181
16	17845	17836	9	1189
17	19022	19025	-3	1177
18	20225	20214	11	1203
19	21386	21403	-17	1161
20	22573	22592	-19	1187
21	23777	23781	-4	1204
22	24981	24971	10	1204
23	26167	26160	7	1186
24	27351	27349	2	1184
25	28531	28538	-7	1180
26	29719	29727	-8	1188
27	30921	30916	5	1202
28	32105	32105	0	1184

Table 1

Analysis of Vatell Sensor Output - Signal Processor Data

Turbine Speed is 12,645 RPM

Blade Number	T-O-A	Ideal Time	Time Error	Delta Time
1	0			
2	1051	1059	-8	1051
3	2110	2118	-8	1059
4	3159	3178	-19	1049
5	4216	4237	-21	1057
6	5289	5296	-7	1073
7	6347	6355	-8	1058
8	7400	7415	-15	1053
9	8469	8474	-5	1069
10	9531	9533	-2	1062
11	10588	10592	-4	1057
12	11636	11651	-15	1048
13	12696	12711	-15	1060
14	13772	13770	2	1076
15	14836	14829	7	1064
16	15893	15888	5	1057
17	16946	16948	-2	1053
18	18011	18007	4	1065
19	19067	19066	1	1056
20	20116	20125	-9	1049
21	21185	21184	1	1069
22	22246	22244	2	1061
23	23308	23303	5	1062
24	24371	24362	9	1063
25	25433	25421	12	1062
26	26484	26481	3	1051
27	27545	27540	5	1061
28	28599	28599	0	1054

Table 2

Analysis of Vatell Sensor Output - Signal Processor Data

Turbine Speed is 14,454.77 RPM

Blade Number	T-O-A	Ideal Time	Time Error	Delta Time
1	0			
2	922	926	-4	922
3	1843	1853	-10	921
4	2771	2779	-8	928
5	3697	3705	-8	926
6	4622	4632	-10	925
7	5559	5558	1	937
8	6477	6485	-8	918
9	7410	7411	-1	933
10	8335	8337	-2	925
11	9263	9264	-1	928
12	10188	10190	-2	925
13	11113	11116	-3	925
14	12042	12043	-1	929
15	12974	12969	5	932
16	13890	13896	-6	916
17	14819	14822	-3	929
18	15753	15748	5	934
19	16671	16675	-4	918
20	17601	17601	0	930
21	18538	18527	11	937
22	19450	19454	-4	912
23	20385	20380	5	935
24	21313	21307	6	928
25	22233	22233	0	920
26	23161	23159	2	928
27	24084	24086	-2	923
	25012	25012	0	928

Table 3

blade number less one. The maximum difference between "ideal" time and measured time for each of the three speeds is 1.7%, 2.0% and 1.2% of the blade to blade interval. Arrival times are most accurate for the highest speed, as might be expected. The effect of a 10 times improvement in signal-to-noise ratio, easily feasible with proper amplifier design, should be to bring these within one count of the proper value. This would result in a timing precision of 0.16 microseconds, representing less than 0.002" tangential displacement of the blades.

The clearance numbers calculated and displayed on the monitor (Figures 12, 13 and 14) are in error for three known reasons. First, the large amount of noise on the signal interferes with peak detection. Because an A/D conversion cycle may be triggered ahead of the waveform peak, the voltage sampled and converted could be grossly wrong. A noise pulse at the peak would not change the timing, but would affect the amplitude measurement.

The second source of error is in the handling of analog to digital results by the TMS 32010 software. While the clearance numbers displayed on the screen show five significant figures, only a few different values are represented. By back-calculating convertor values from the results shown on the screen for 14,454.77 RPM, this error can be localized.

Indicated Clearance	Calculated A/D Value	Equivalent A/D 12 bits
33.08344	592	0010 0101 0000
31.38583	624	0010 0111 0000
29.85384	656	0010 1001 0000
28.46436	688	0010 1011 0000
27.19839	720	0010 1101 0000
26.04016	752	0010 1111 0000
24.97648	784	0011 0001 0000
23.99622	816	0011 0011 0000

From this analysis it can be seen that the 5 least significant bits of convertor data are not changing. The reason for this is unknown. The analog to digital convertor was tested independently, and produced the full range of output numbers. The complementing and shifting routine for A/D data was also debugged and checked thoroughly. The cause of the problem may be noise affecting the sample and hold output.

The third source of error is the use of incorrect values for the constants in the clearance calculation. Values for the constants A, B, C and D were written into the program before measurements and analysis of the sensor actually used in the tests. These contants were not corrected in the program because time ran out for the experiment: a great deal more time was consumed in troubleshooting the system than anticipated, most of it with the engine running.

Clearances can be calculated using the known constants A, B, C and D for the sensor used in these tests, but the clearance for one blade must be known (or assumed) to determine the value for the constant K_s in the program. If it is assumed that the highest signal amplitude measured at 14,454.77 RPM represents a physical clearance of .025 inches, then K_s = .03353, and the A/D measurements for that speed would represent a range of clearances from .025 inches to .043 inches.

CONCLUSIONS

Despite experimental problems with amplifier noise and the necessity to perform some calculations off-line, this project has proven the feasibility of correcting the Vatell blade clearance sensor output signal for speed effects. A peak sensing technique with time resolution of 0.16 microsecond has been demonstrated. With achievable noise reductions in the analog signal path this method should yield a tangential resolution of .002 inches, sufficient for the detection of blade vibration or twist at full speed on the example engine. With correct handling of A/D conversion results and the use of proper constants for the sensor being employed, it should be possible to achieve on-line speed compensation with a computed clearance precision of .0001 inches or better.

A double precision divide is essential to on-line calculation of clearance, so the selection of a microprocessor for future systems should take this requirement into account.

A great deal of time and effort could have been saved in troubleshooting and calibrating this system if an accurate electrical simulation for the once-per-revolution and blade signals had been arranged. It is impractical to debug software and find digital and analog signal processing hardware problems under the time and cost constraints of jet engine test cell operation.

FUTURE RESEARCH

The next step in this research should be to build a production prototype signal processing system with an improved first stage amplifier, using a microprocessor which will perform the required calculations on-line. With proper attention to the test and communications features of the new system, the gathering of data on compressor blade clearance, vibration and twist should become practical in a wide variety of circumstances. For lowest cost of development and use, the new system should be designed as a microcomputer plug-in. This strategy will open up many possibilities for turbomachinery data acquisition and analysis at reasonable capital cost and with application flexibility. Compressor blade clearance control, vibration monitoring and similar uses of the data from Vatell sensors should then proceed. As high temperature versions of the sensor are developed, similar applications will become possible in turbines. The benefits of casing-based monitoring of turbomachinery blade conditions will then be broadly available and economical to acquire.

For convenience and economy in tests of the new system, an accurate electrical simulator for sensor and once-per revolution signals should be constructed. This simulator should have the flexibility to adjust the amplitudes and timing of individual signals, so that response of the tip clearance signal processor to anticipated signals can be determined before trials in the test cell environment.

A mechanical mockup of a turbomachine blade row with known variations in end gap should be constructed, preferably in a temperature-controlled chamber. This apparatus would be used to calibrate sensors and test signal processors, and to explore the effects of temperature on sensor calibration and operation.

The potential of blade vibration and twist monitoring should be explored using two or more sensors, together with processors whose timing is slaved or coordinated. One objective for such a combination would be to detect rotating stall. This should be explored experimentally in a compressor research facility.

REFERENCES

(1) Wilson, D. S., "An Investigation of Sensors Suitable for Monitoring Blade Deflections for a VA1310 Wind Tunnel Compressor" - AFWAL-TR-81-3076 Final Report on Contract F33615-79-C-3019 July 1981

(2) Kiraly, L. J., "Digital System for Dynamic Turbine Engine Blade Displacements" - Measurement Methods on Rotating Components of Turbomachinery, ASME Gas Turbine Symposium, New York, 1980

(3) Wilson, D. S., "Compressor Blade Monitoring System for a VA1310 (Allis Chalmers) Wind Tunnel Compressor" - Final Report AFWAL Contract No. F33615-79-C-3019 July 1980

(4) Barranger, J. P. and Ford, M. J., "Laser Optical Blade Tip Clearance Measurement System" - Measurement Methods on Rotating Components of Turbomachinery, ASME Gas Turbine Symposium, New York, 1980

(5) Roth, H., "Vibration and Clearance Measurements on Rotating Blades Using Stationary Probes" - Measurement Techniques in Turbomachines, Von Karman Institute For Fluid Dynamics Lecture Series, May 18-22, 1981

(6) O'Brien, W. F., Sparks, J. F. and Dellinger, D. F., "Non-Contacting Method for Measurement of Dynamic Blade Motions in Axial-Flow Compressors" - Proceedings of the 27th International Instrumentation Symposium, ISA, Indianapolis, IN, April 1981

```
10  REM
20  REM       BASIC INTERFACE PROGRAM                       HKK  5/9/88
30  REM THIS BASIC PROGRAM INTERFACES WITH THE ASPI DSP BOARD SOFTWARE.
40  REM IT WAITS FOR A KEY INPUT FROM THE PC KEYBOARD.  WHEN IT GETS ONE
50  REM IT SETS THE DOWNLOAD FLAG IN THE ASPI'S PROGRAM MEMORY.
60  REM WHEN THE 0 BLADE HAS PASSED, THEN THE 32010 SOFTWARE WILL STORE
70  REM ALL CURRENT TIME-OF-ARRIVAL VALUES AND A/D'ED AMPLITUDE VALUES
80  REM IN THE PROGRAM MEMORY.  THE 32010 WILL THEN RESET THE DOWNLOAD
90  REM FLAG INDICATING TO THE PC THAT THE DATA IS PRESENT AND CAN BE READ.
100 REM THE PC WILL THEN READ THE DATA, CALCULATE THE GAP CLEARANCES,
110 REM AND DISPLAY THE INFO FOR ALL 28 BLADES.
150 REM
151     K1 = 1.2E-08
152     K2 = .0001
153     K3 = .31
154     K4 = .02
155 REM K5 = 7.0E-05
156     INPUT "Enter K5, A/D output multplier - ", K5
159     DEF SEG = 40960
160     CLS
170     LOCATE 22, 1
180     PRINT "Press any key to DOWNLOAD DATA..."
190 REM LOOP HERE UNTIL A KEY IS PRESSED
200     V$ = INKEY$
210     IF V$ = "" THEN 200
215     IF V$ = "S" THEN 495
220     CLS
230     LOCATE 3, 1
293     POKE 256, 1: POKE 257, 0
295     IF PEEK(256) + (256! * PEEK(257)) = 0 THEN 300
297     GOTO 295
300     SPDCNT = PEEK(370) + (256! * PEEK(371))
305     IF SPDCNT = 0 THEN SPEED = 0: GOTO 320
310     SPEED = 60 * (1 / (SPDCNT * 1.6E-07))
320     PRINT "TURBINE SPEED IS "; SPEED; " RPM"
330     PRINT ""
350     PRINT "      T-O-A", "GAP-IN./1000", , "      T-O-A", "GAP-IN./1000"
360     PRINT ""
370 REM LOOP WHERE VALUES ARE CALCULATED AND DISPLAYED
380     FOR I = 1 TO 14
390             TOA = PEEK(312 + (I * 2)) + (256! * PEEK(313 + (I * 2)))
400             AD = PEEK(256 + (I * 2)) + (256! * PEEK(257 + (I * 2)))
410             IF AD = 0 THEN GAP = 0: GOTO 430
420             GAP = ((K1 * SPEED ^ 2) - (K2 * SPEED) + K3) / (AD * K5) - K4
430             TOA2 = PEEK(340 + (I * 2)) + (256! * PEEK(341 + (I * 2)))
440             AD2 = PEEK(284 + (I * 2)) + (256! * PEEK(285 + (I * 2)))
450             IF AD2 = 0 THEN GAP2 = 0: GOTO 470
460             GAP2 = ((K1 * SPEED ^ 2) - (K2 * SPEED) + K3) / (AD2 * K5) - K4
463             IF I > 9 THEN 470
465             PRINT " "; I; "- "; TOA, GAP, , I + 14; "- "; TOA2, GAP2: GOTO
480
470             PRINT I; "- "; TOA, GAP, , I + 14; "- "; TOA2, GAP2
480     NEXT I
490     GOTO 170
```

Appendix A

```
10  REM
20  REM        BASIC INTERFACE PROGRAM                          HKK  5/9/88
30  REM THIS BASIC PROGRAM INTERFACES WITH THE ASPI DSP BOARD SOFTWARE.
40  REM IT WAITS FOR A KEY INPUT FROM THE PC KEYBOARD.  WHEN IT GETS ONE
50  REM IT SETS THE DOWNLOAD FLAG IN THE ASPI'S PROGRAM MEMORY.
60  REM WHEN THE 0 BLADE HAS PASSED, THEN THE 32010 SOFTWARE WILL STORE
70  REM ALL CURRENT TIME-OF-ARRIVAL VALUES AND A/D'ED AMPLITUDE VALUES
80  REM IN THE PROGRAM MEMORY.  THE 32010 WILL THEN RESET THE DOWNLOAD
90  REM FLAG INDICATING TO THE PC THAT THE DATA IS PRESENT AND CAN BE READ.
100 REM THE PC WILL THEN READ THE DATA, CALCULATE THE GAP CLEARANCES,
110 REM AND DISPLAY THE INFO FOR ALL 28 BLADES.
150 REM
151     K1 = 1.2E-08
152     K2 = .0001
153     K3 = .31
154     K4 = .02
155 REM K5 = 7.0E-05
156     INPUT "Enter K5, A/D output multplier - ", K5
159     DEF SEG = 40960
160     CLS
170     LOCATE 22, 1
180     PRINT "Press any key to DOWNLOAD DATA..."
190 REM LOOP HERE UNTIL A KEY IS PRESSED
200     V$ = INKEY$
210     IF V$ = "" THEN 200
215     IF V$ = "S" THEN 495
220     CLS
230     LOCATE 3, 1
293     POKE 256, 1: POKE 257, 0
295     IF PEEK(256) + (256! * PEEK(257)) = 0 THEN 300
297     GOTO 295
300     SPDCNT = PEEK(370) + (256! * PEEK(371))
305     IF SPDCNT = 0 THEN SPEED = 0: GOTO 320
310     SPEED = 60 * (1 / (SPDCNT * 1.6E-07))
320     PRINT "TURBINE SPEED IS "; SPEED; " RPM"
330     PRINT ""
350     PRINT "      T-O-A", "GAP-IN./1000", , "      T-O-A", "GAP-IN./1000"
360     PRINT ""
370 REM LOOP WHERE VALUES ARE CALCULATED AND DISPLAYED
380     FOR I = 1 TO 14
390             TOA = PEEK(312 + (I * 2)) + (256! * PEEK(313 + (I * 2)))
400             AD = PEEK(256 + (I * 2)) + (256! * PEEK(257 + (I * 2)))
410             IF AD = 0 THEN GAP = 0: GOTO 430
420             GAP = ((K1 * SPEED ^ 2) - (K2 * SPEED) + K3) / (AD * K5) - K4
430             TOA2 = PEEK(340 + (I * 2)) + (256! * PEEK(341 + (I * 2)))
440             AD2 = PEEK(284 + (I * 2)) + (256! * PEEK(285 + (I * 2)))
450             IF AD2 = 0 THEN GAP2 = 0: GOTO 470
460             GAP2 = ((K1 * SPEED ^ 2) - (K2 * SPEED) + K3) / (AD2 * K5) - K4
463             IF I > 9 THEN 470
465             PRINT " "; I; "- "; TOA, GAP, , I + 14; "- "; TOA2, GAP2: GOTO
480
470             PRINT I; "- "; TOA, GAP, , I + 14; "- "; TOA2, GAP2
480     NEXT I
490     GOTO 170
```

Appendix A

```
           *
           * This TMS32010 Assembly Language Program allows the DSP
           * board from ATLANTA SIGNAL PROCESSOR to work in con-
           * junction with the Signal Interface Board to monitor the
           * speed and signal amplitude information from the VATELL
           * turbine sensor.  The program is interrupt-driven and
           * reads and stores required information for downloading to
           * the PC host.  A BASIC program in the PC takes the infor-
           * mation and calculates a blade-to-sensor clearance based
           * upon sensor signal amplitude and time-of-arrival data.
           * This particular Signal Interface board and software pro-
           * gram is designed to operate with a turbine having 28
           * blades and operating up to a maximum speed of 16,000 RPM.
           * Board timing is based upon a processor clock speed of 25
           * MHz.
           *
           * SYMBOLS - DATA MEMORY LOCATIONS USED *
           *
0002       STATUS  EQU  >2            STORE STATUS DURING INTRPT.
0003       ACH     EQU  >3            STORE ACC. HIGH DURING INT.
0004       ACL     EQU  >4            STORE ACC. LOW  "       "
0005       AR00    EQU  >5            STORE AUX. REG. 0 "     "
0006       AR01    EQU  >6            STORE AUX. REG. 1 "     "
0008       BLDPTR  EQU  >8            BLADE POSITION POINTER
0009       INITFG  EQU  >9            INITIALIZING FLAG
000A       CLCFLG  EQU  >A            FLAG- START CALCULATION
000B       SPEED   EQU  >B            SPEED OF PREV. ROTATION
000C       CUMLTV  EQU  >C            CUMULATIVE TOTAL OF TIME
000D       PRVNDX  EQU  >D            PREV. INDEX VALUE
000E       ONEFLG  EQU  >E            ONE TIME AROUND FLAG
000F       DA      EQU  >F
0010       LOPCTR  EQU  >10
0011       CNTRED  EQU  >11           STORE READING OF COUNTER
0012       INDEX   EQU  >12           CURRENT INDEX READING
0013       TEMP1   EQU  >13           TEMP. STORAGE LOCATIONS
0014       TEMP2   EQU  >14
0015       TEMP3   EQU  >15
0016       ADRED   EQU  >16           TEMP. STORAGE OF A/D VALUE
           *
002B       AD0     EQU  >2B           A/D VALUE - 1ST BLADE AFTER
           *               INDEX. 2BH THROUGH 46H CONTAIN A/D VALUES
           *               FOR 28 BLADES
0047       TOA0    EQU  >47           TIME OF ARRIVAL - 1ST BLADE
           *               AFTER INDEX. 47H THROUGH 62H CONTAIN TIME
           *               OF ARRIVAL FOR 28 BLADES
0063       CNTRD0  EQU  >63           COUNTER VALUE WHEN BLADE
           *               ARRIVES
           *
```

Appendix A

```
0051                *
0052                * INITIAL JUMP VECTORS AND CONSTANTS STORED IN FIRST PAGE
0053                * OF PROGRAM MEMORY
0054                *
0055 0000                   AORG   0
0056                *
0057 0000 F900              B      INIT
     0001 0100
0058 0002 F900              B      INTRPT
     0003 01C2
0059                *
0060 0010                   AORG   >10
0061 0010 0000      CNSNTS  DATA   0,1,0,0,0,1,0,>0A00,0
     0011 0001
     0012 0000
     0013 0000
     0014 0000
     0015 0001
     0016 0000
     0017 0A00
     0018 0000
0062                *  CONSTANTS FOR CALCULATIONS
0063 0019 FFFF      CLCNTS  DATA   >FFFF,>8000,>7FFF,>1000,>1000
     001A 8000
     001B 7FFF
     001C 1000
     001D 1000
0064 0080                   AORG   >80
0065 0080 0000      DWNLOD  DATA   0               FLAG LOCATION FOR TALKING
0066                *                  WITH PC.  IF SET TO 1,  PC WANTS DATA
0067                *                  DOWNLOADED.  RESET WHEN DONE.
0068                *                              LOCATIONS FOR STORING A/D
0069                *                  VALUES AND TIME OF ARRIVAL VALUES FOR
0070                *                  READING BY PC.
0071 0081 0000      AD      DATA   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     0082 0000
     0083 0000
     0084 0000
     0085 0000
     0086 0000
     0087 0000
     0088 0000
     0089 0000
     008A 0000
     008B 0000
     008C 0000
     008D 0000
     008E 0000
     008F 0000
     0090 0000
     0091 0000
```

```
      0092 0000
      0093 0000
      0094 0000
      0095 0000
      0096 0000
 0072 0097 0000          DATA    0,0,0,0,0,0
      0098 0000
      0099 0000
      009A 0000
      009B 0000
      009C 0000
 0073 009D 0000  TOA     DATA    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      009E 0000
      009F 0000
      00A0 0000
      00A1 0000
      00A2 0000
      00A3 0000
      00A4 0000
      00A5 0000
      00A6 0000
      00A7 0000
      00A8 0000
      00A9 0000
      00AA 0000
      00AB 0000
      00AC 0000
      00AD 0000
      00AE 0000
      00AF 0000
      00B0 0000
      00B1 0000
      00B2 0000
 0074 00B3 0000          DATA    0,0,0,0,0,0
      00B4 0000
      00B5 0000
      00B6 0000
      00B7 0000
      00B8 0000
 0075 00B9 0000  SPED    DATA    0
 0076 00BA 0001  TOOSLO  DATA    1           FLAG TELLS PC RUNNING SLOW
 0077            *
```

Appendix A

```
                   * INIT - INITIALIZATION PART OF PROGRAM
                   *       PROGRAM INITIALIZATION BEGINS HERE. ALL REGISTERS
                   *       ARE INITIALIZED, CONSTANTS TRANSFERRED FROM PROG.
                   *       MEMORY TO DATA MEMORY
0100                       AORG    >100            START PROGRAM ON PAGE 1
0100 7F81          INIT    DINT
0101 7F8B                  SOVM                    SET OVERFLOW MODE
0102 6880                  LARP    0
0103 6E00                  LDPK    0
0104 F800                  CALL    NITLIZ          CALL SUBROUTINE TO INIT.
0105 014A
                   *                               PRIMARY REGISTERS
0106 7137                  LARK    AR1,55          PUT IN REG. 1 # OF VALUES
0107 7062                  LARK    AR0,>62         PUT IN REG. 0 TOP ADDRESS
                   *                         OF DATA MEMORY
0108 6880          LOPDAT  LARP    0               MAKE SURE ARP = 0
0109 3113                  SAR     1,TEMP1         STORE CURRENT COUNT. VALUE
010A 7E81                  LACK    >81             BASE ADDRESS OF DATA IN PROG.
010B 6113                  ADDS    TEMP1           CALC. PROG. ADDRESS
010C 6791                  TBLR    *-,1          READ DATA, STORE @ADDRESS
                   *                         POINTED AT, DEC. AR0, CHANGE ARP=1
010D F400                  BANZ    LOPDAT          IF AR1 NOT 0, GO BACK
010E 0108
010F 6880                  LARP    0               RESET AUX.REG.PTR.
0110 4E0F                  OUT     DA,PA6          SET VALID D/A LEVEL
0111 7E01                  LACK    1
0112 5013                  SACL    TEMP1
0113 4913                  OUT     TEMP1,PA1       SET ENABLE2 SO S/H WILL SAMP.
0114 7E00                  LACK    0               SET A/D INPUT LOW
0115 5013                  SACL    TEMP1
0116 4A13                  OUT     TEMP1,PA2
0117 F900                  B       MAIN            GO TO MAIN LOOP
0118 0130
                   *
```

```
                * MAIN - MAIN PROGRAM LOOP
                *
0130                   AORG    >130
0130 660A  MAIN        ZALS    CLCFLG           SEE IF NEED TO GO DO CALC.
0131 FF00              BZ      NOCALC           IF FLAG=0, DON'T CALL SUB.
0132 0135
0133 F800              CALL    CALC             IF FLAG=1, GO FOR IT
0134 0155
0135 7E80  NOCALC      LACK    >80              ADDRESS OF DOWNLOAD FLAG
0136 6713              TBLR    TEMP1
0137 6613              ZALS    TEMP1
0138 FF00              BZ      NODNLD
0139 013F
013A 6608              ZALS    BLDPTR
013B FE00              BNZ     NODNLD
013C 013F
013D F800              CALL    DWNLD
013E 0196
013F 4113  NODNLD      IN      TEMP1,PA1
0140 7E02              LACK    2
0141 7913              AND     TEMP1
0142 FE00              BNZ     NODNLD
0143 013F
0144 7E01              LACK    1
0145 5013              SACL    TEMP1
0146 4813              OUT     TEMP1,PA0
0147 7F82              EINT
0148 F900  LOOPIT      B       LOOPIT           GO AROUND AGAIN
0149 0148
                *
```

```
                * SUBROUTINE NITLIZ - INITIALIZE
                *       THIS SUBROUTINE CAN BE CALLED FROM ANYWHERE IN
                *       THE PROGRAM. IT WILL INITIALIZE THE NECESSARY
                *       DATA MEMORY LOCATIONS FROM VALUES IN THE PROGRAM
                *       MEMORY. THIS IS DONE PRIOR TO STARTING TO READ
                *       FROM SCRATCH. IT WOULD BE NECESSARY TO CALL IT IF
                *       THE TURBINE IS RUNNING TOO SLOW TO GET PROPER
                *       VALUES.
014A 7107       NITLIZ  LARK    AR1,>7          PUT IN REG. 1 # OF VALUES
014B 700F               LARK    AR0,>0F         PUT IN REG. 0 TOP ADDRESS
                *                            OF DATA MEMORY
014C 6880       MOVDAT  LARP    0               MAKE SURE ARP = 0
014D 3113               SAR     1,TEMP1         STORE CURRENT COUNT. VALUE
014E 7E10               LACK    >10             BASE ADDRESS OF DATA IN PROG.
014F 6113               ADDS    TEMP1           CALC. PROG. ADDRESS
0150 6791               TBLR    *-,1          READ DATA, STORE @ADDRESS
                *                            POINTED AT, DEC. AR0, CHANGE ARP=1
0151 F400               BANZ    MOVDAT          IF AR1 NOT 0, GO BACK
0152 014C
0153 6880               LARP    0               RESET AUX.REG.PTR.
0154 7F8D               RET
                *
```

```
                * CALC SUBROUTINE - SPEED ACCUMULATION AND CALCULATIONS
                *      ARE DONE HERE.  SET OR RESET TOOSLO FLAG.
                *
0155 6608 CALC   ZALS  BLDPTR          GET POINTER TO T-O-A VALUES
0156 FF00        BZ    ISOBLD          IF BLDPTR = 0, BRANCH
0157 016A
0158 6880        LARP  0               SET AUX. REG. PTR. = 0
0159 3808        LAR   0,BLDPTR        FETCH INDEX TO T-O-A NOS.
015A 6898        MAR   *-              DECREMENT REGISTER TO PREV.
                *                      T-O-A LOCATION
015B 3013        SAR   0,TEMP1         STORE TEMPORARILY
015C 7E63        LACK  >63             CALC. ADDRESS TO PUT T-O-A
015D 6113        ADDS  TEMP1           ADD PTR. TO 1ST LOC. ADDRESS
015E 5013        SACL  TEMP1           STORE TEMPORARILY
015F 3813        LAR   0,TEMP1         NOW PUT CALC. ADD. IN AR
0160 6688        ZALS  *               FETCH TIME OF ARRIVAL DATA
0161 5013        SACL  TEMP1           STORE INFO. TEMP.
0162 7E63        LACK  >63
0163 6108        ADDS  BLDPTR
0164 5014        SACL  TEMP2
0165 3814        LAR   0,TEMP2
0166 6688        ZALS  *               FETCH LAST BLADE'S T-O-A
0167 6313        SUBS  TEMP1           SUBTRACT LAST T-O-A FROM PREV
0168 F900        B     AROUND          GO AROUND NEXT. FOR 0 BLADE.
0169 016C
016A 6663 ISOBLD ZALS  CNTRD0          FETCH T-O-A VALUE FOR 0 BLADE
016B 637E        SUBS  CNTRD0+27       SUBTRACT T-O-A VALUE FOR 27
                *                      BLADE
                *  IF WE'VE GOTTEN HERE, WE HAVE THE ELAPSED TIME BETWEEN
                *  THE LAST T-O-A AND THE PRESENT IN THE ACCUMULATOR
016C 610C AROUND ADDS  CUMLTV          ADD TO CUMULATIVE TOTAL
016D 500C        SACL  CUMLTV          STORE BACK IN REG.
                *  NOW INITIALIZE, SET INITFLG, ENABLE SAMPLE, AND TRY AGAIN
                *  WOW! WE FINALLY GOT HERE. NOW WHAT? LET'S SEE, OH YEAH,
                *  WE CAN NOW TEST FOR ONEFLG. IF SET, ENABLE SAMPLE AND GO
                *  BACK.
016E 7E47 ACCISO LACK  >47
016F 6108        ADDS  BLDPTR          STORE ACCUMULATED TIME IN
0170 5013        SACL  TEMP1           T-O-A DATA LOCATION
0171 3813        LAR   0,TEMP1
0172 660C        ZALS  CUMLTV          FETCH DATA
0173 5088        SACL  *
0174 6608        ZALS  BLDPTR          IF 0 BLADE, SPEED=CUMLTV &
0175 FE00        BNZ   NTOBLD          CUMLTV=0
0176 017E
0177 660C        ZALS  CUMLTV
0178 500B        SACL  SPEED
0179 7EB9        LACK  >B9             WRITE SPEED TO PROGRAM MEMORY
017A 7D0B        TBLW  SPEED
017B 7E00        LACK  0
017C 500C        SACL  CUMLTV
```

```
017D 5047           SACL    TOAO            SET TOA FOR 0 BLADE=0
                  * SEE IF A/D DONE BEFORE READING
017E F600  NTOBLD   BIOZ    ADDONE          IF BIO LOW, A/D IS DONE
017F 0182
0180 F900           B       NTOBLD          IF HIGH, GO AROUND UNTIL DONE
0181 017E
0182 4616  ADDONE   IN      ADRED,PA6       READ A/D CONVERTER
0183 7E01           LACK    1
0184 5013           SACL    TEMP1           CAN RELEASE SAMPLE/HOLD NOW
0185 4913           OUT     TEMP1,PA1       GOT IT
0186 7E00           LACK    0               RESET CALCULATION FLAG
0187 500A           SACL    CLCFLG
                  * FIRST HAVE TO GET A/D VALUE IN FORM WE CAN USE IN CALC.
0188 7E19           LACK    >19
0189 6713           TBLR    TEMP1           FETCH FFFF MASK FOR XOR'ING
018A 6613           ZALS    TEMP1
018B 7816           XOR     ADRED           DO 1'S COMPLEMENT
018C 5016           SACL    ADRED
018D 2C16           LAC     ADRED,12        SHIFT RIGHT 4 TIMES
018E 5816           SACH    ADRED
                  * STORE THIS NUMBER IN INDEXED FILE FOR A/D VALUES
018F 7E2B           LACK    >2B
0190 6108           ADDS    BLDPTR          DEVELOP INDEX FROM BLADE NO.
0191 5013           SACL    TEMP1
0192 3813           LAR     0,TEMP1         PUT INDEX INTO AUX. REG.
0193 6616           ZALS    ADRED
0194 5088           SACL    *               STORE IT NOW
0195 7F8D           RET                     ALL DONE, GO BACK NOW
                  *
```

```
                * DOWNLOAD SUBROUTINE - DOWNLOAD TIME OF ARRIVAL AND
                *   CLEARANCE INFORMATION TO PC.  THE DOWNLOAD FLAG HAS BEEN
                *   SET, THAT'S WHY WE'RE HERE.  TRANSFER T-O-A'S AND A/D
                *   DATA TO PROG. MEMORY, RESET DOWNLOAD FLAG AND RET
                *
0196 7137       DWNLD   LARK    AR1,55          PUT IN REG. 1 # OF VALUES
0197 7062               LARK    AR0,>62         PUT IN REG. 0 TOP ADDRESS
                *                          OF DATA MEMORY
0198 6880       LPDAT   LARP    0               MAKE SURE ARP = 0
0199 3113               SAR     1,TEMP1         STORE CURRENT COUNT. VALUE
019A 7E81               LACK    >81             BASE ADDRESS OF DATA IN PROG.
019B 6113               ADDS    TEMP1           CALC. PROG. ADDRESS
019C 7D91               TBLW    *-,1            WRITE DATA TO PROG. ADDRESS
                *                          POINTED AT, DEC. AR0, CHANGE ARP=1
019D F400               BANZ    LPDAT           IF AR1 NOT 0, GO BACK
019E 0198
019F 6880               LARP    0               RESET AUX.REG.PTR.
01A0 7EB9               LACK    >B9             STORE SPEED READING
01A1 7D0B               TBLW    SPEED
01A2 7E00               LACK    0
01A3 5013               SACL    TEMP1           RESET DOWNLOAD FLAG
01A4 7E80               LACK    >80
01A5 7D13               TBLW    TEMP1
01A6 7F8D               RET
                *
```

```
                * CLEAR SUBROUTINE - CLEARS TIME OF ARRIVAL AND CLEARANCE
                *   VALUES IN DATA MEMORY. CALLED WHEN INITIALIZING AND
                *   WHEN LOOP COUNTER DECREMENTS OUT.
                *
01A7 7137  CLEER   LARK   AR1,55         PUT IN REG. 1 # OF VALUES
01A8 7062          LARK   AR0,>62        PUT IN REG. 0 TOP ADDRESS
                *                    OF DATA MEMORY
01A9 6880  LPDAT1  LARP   0              MAKE SURE ARP = 0
01AA 3113          SAR    1,TEMP1        STORE CURRENT COUNT. VALUE
01AB 7E81          LACK   >81            BASE ADDRESS OF DATA IN PROG.
01AC 6113          ADDS   TEMP1          CALC. PROG. ADDRESS
01AD 6791          TBLR   *-,1         READ DATA, STORE @ADDRESS
                *                    POINTED AT, DEC. AR0, CHANGE ARP=1
01AE F400          BANZ   LPDAT1         IF AR1 NOT 0, GO BACK
01AF 01A9
01B0 6880          LARP   0              RESET AUX.REG.PTR.
01B1 7F8D          RET
                *
```

```
                * CLRPRG - CLEAR PROGRAM MEMORY SUBROUTINE
                *   THIS SUB. CLEARS THE T-O-A AND CLEARANCE VALUES IN THE
                *   PROGRAM MEMORY LOCATIONS TO 0.  DONE WHEN SPEED TOO SLOW
                *
01B2 7E00  CLRPRG  LACK   0                0 INTO ACC.
01B3 5014          SACL   TEMP2            STORE IN TEMPORARY REGISTER
01B4 7138          LARK   AR1,56           PUT IN REG. 1 # OF VALUES
01B5 6881  LPDAT2  LARP   1                MAKE SURE ARP = 1
01B6 3113          SAR    1,TEMP1          STORE CURRENT COUNT. VALUE
01B7 7E81          LACK   >81              BASE ADDRESS OF DATA IN PROG.
01B8 6113          ADDS   TEMP1            CALC. PROG. ADDRESS
01B9 7D14          TBLW   TEMP2          READ DATA, STORE @ADDRESS
                *                      POINTED AT, DEC. AR1
01BA F400          BANZ   LPDAT2           IF AR1 NOT 0, GO BACK
01BB 01B5
01BC 6880          LARP   0                RESET AUX.REG.PTR.
01BD 7E01          LACK   1
01BE 5013          SACL   TEMP1            SET TOOSLO FLAG FOR PC
01BF 7EBA          LACK   >BA
01C0 7D13          TBLW   TEMP1
01C1 7F8D          RET
                *
```

```
                *
                *
 01C2 7C02 INTRPT SST     STATUS        1  STORE STATUS ON PAGE 1
 01C3 6E01        LDPK    1             1  SET DATA PAGE = 1
 01C4 5803        SACH    ACH           1  STORE ACCUMULATOR
 01C5 5804        SACH    ACL           1
 01C6 3005        SAR     0,AR00        1  STORE AUX. REGISTERS
 01C7 3106        SAR     1,AR01        1
 01C8 6E00        LDPK    0             1  DATA PAGE PTR. = 0
                *
 01C9 7E00        LACK    0             1  HURRY UP AND HOLD S/H SO
                *                          WON'T LOSE DATA
 01CA 5013        SACL    TEMP1         1
 01CB 4913        OUT     TEMP1,PA1     2  GOT IT
 01CC 4813        OUT     TEMP1,PA0        FIX SO CAN HAVE NO MORE INT.
 01CD 7E01        LACK    1                TO SETTLE
 01CE 5013        SACL    TEMP1            PULSE THE A/D TO START IT
 01CF 4A13        OUT     TEMP1,PA2
 01D0 7E00        LACK    0
 01D1 5013        SACL    TEMP1
 01D2 4A13        OUT     TEMP1,PA2        480 nSEC PULSE
 01D3 4011        IN      CNTRED,PA0       GET TIME-OF-ARRIVAL DATA
 01D4 6880        LARP    0
                *
                * THIS IS A REGULAR TRIP THROUGH THE INTERRUPT ROUTINE
                *
 01D5 7E01        LACK    1                INCREMENT BLADE POINTER
 01D6 6108        ADDS    BLDPTR
 01D7 5008        SACL    BLDPTR           PUT BACK IN REGISTER
 01D8 660D        ZALS    PRVNDX           TEST TO SEE IF THE INDEX AT
                *                          LAST INTERRUPT WAS A 0
 01D9 FE00        BNZ     PNDXNO           IF NOT, GO ON
 01DA 01E7
                * IF YOU GOT HERE, PREVIOUS INDEX WAS 0
 01DB 4112        IN      INDEX,PA1        FETCH INDEX INFO. TO SEE IF
                *                          INDEX IS PRESENTLY A 0
 01DC 7E01        LACK    1                MASK ALL BUT LSB
 01DD 7912        AND     INDEX            AND WITH ACC.
 01DE 5012        SACL    INDEX
 01DF 500D        SACL    PRVNDX           MAKE PRVNDX = INDEX
 01E0 FF00        BZ      NDXSLO           IF INDEX STILL 0, GO ON
 01E1 01EC
                * INDEX HAS CHANGED FROM 0 TO 1, THEREFORE SET BLADE PTR.
                * TO 0 AND GO ON
 01E2 7E00        LACK    0
 01E3 5008        SACL    BLDPTR
 01E4 500E        SACL    ONEFLG           RESET ONE TIME AROUND FLAG
                *                          IF SET
 01E5 F900        B       NDXSLO           GO AROUND NEXT BECAUSE IT'S
 01E6 01EC
                *       Appendix A         FOR PRVNDX = 1
```

```
0316 01E7 4112  PNDXNO  IN     INDEX,PA1    SET PRVNDX = INDEX, WHETHER
0347 01E8 7E01          LACK   1            INDEX HAS CHANGED FROM 1 TO
0348 01E9 7912          AND    INDEX        0 OR NOT
0349 01EA 5012          SACL   INDEX
0350 01EB 500D          SACL   PRVNDX
0351 01EC 7E01  NDXSLO  LACK   1            SET CALC. FLAG SO WE WILL
0352 01ED 500A          SACL   CLCFLG       KNOW TO DO THEM
0353            *
0354            *
0355 01EE 7E1C  RESTOR  LACK   28           TEST TO SEE IF GONE OVER 27
0356 01EF 6308          SUBS   BLDPTR       BLADES DUE TO NOISE
0357 01F0 FE00          BNZ    NOTOVR       IF ACC. NOT 0, NOT OVER
     01F1 01F9
0358 01F2 7E1B          LACK   27
0359 01F3 5008          SACL   BLDPTR
0360 01F4 7E00          LACK   0
0361 01F5 500A          SACL   CLCFLG
0362 01F6 6880          LARP   0
0363 01F7 F900          B      ISOVR
     01F8 0200
0364 01F9 6880  NOTOVR  LARP   0
0365 01FA 7E63          LACK   >63          CALC. ADDRESS TO PUT T-O-A
0366 01FB 6108          ADDS   BLDPTR       ADD PTR. TO 1ST LOC. ADDRESS
0367 01FC 5013          SACL   TEMP1        STORE TEMPORARILY
0368 01FD 3813          LAR    0,TEMP1      NOW PUT CALC. ADD. IN AR
0369 01FE 6611          ZALS   CNTRED       FETCH TIME OF ARRIVAL DATA
0370 01FF 5088          SACL   *            STORE INFO.
0371 0200 7F9D  ISOVR   POP                 PUSH STARTING ADDRESS OF MAIN
0372 0201 7E01          LACK   >1           LOOP ON STACK
0373 0202 5013          SACL   TEMP1
0374 0203 7E30          LACK   >30
0375 0204 0813          ADD    TEMP1,8
0376 0205 7F9C          PUSH
0377            *  CAN GO ON NOW
0378 0206 6E01          LDPK   1            SET DATA PAGE = 1 IF MOD.
0379 0207 3805          LAR    0,AR00       LOAD AUX. REG. WITH OLD VAL.
0380 0208 3906          LAR    1,AR01
0381 0209 6503          ZALH   ACH          LOAD ACC. WITH OLD VALUE
0382 020A 6104          ADDS   ACL          ADD IN LOW ORDER 16 BITS
0383 020B 7B02          LST    STATUS       RESTORE STATUS
0384 020C 6E00          LDPK   0
0385 020D 7F81          DINT                DON'T ENABLE INTERRUPTS YET
0386 020E 7F8D          RET
0387            *
0388                    END
NO ERRORS, NO WARNINGS
^Z
```

Appendix A

CERTIFICATION OF TECHNICAL DATA CONFORMITY

The Contractor, Vatell Corporation, hereby certifies that, to the best of its knowledge and belief, the technical data delivered herewith under Contract No. F33615-87-C-2802 is complete, accuiate, and complies with all requirements of the contract.

24 May 1988
Date

Lawrence W. Langley
Lawrence W. Langley, President

ABSTRACT

A signal processing system was designed and constructed to detect peaks of the Vatell eddy current clearance sensor signal, measure their amplitude and timing, and compute a correction for machine speed to indicate clearance and time of arrival. The system is based on an ADPIBM301 signal processing plug-in for the PC/XT, and consists of a second plug-in, along with special software for signal detection and processing. Tests on a Pratt and Whitney JT15D first stage fan demonstrated timing precision of 0.16 μsecond and indicated individual blade clearances. A capability to perform the speed correction calculation and indicate blade clearance continuously in real time is projected, based on the test results.